User-Network Oriented Call Control and Traffic Management in B-ISDNs

A thesis presented for the degree of
Doctor of Philosophy
in
Electrical and Electronic Engineering
at the
University of Canterbury
Christchurch, New Zealand

August 1994

Vincentius Felix Hartanto
B.E. (Hons I)

To My Dear Parents

ABSTRACT

This thesis presents a framework for call control and traffic management in B-ISDNs. Unlike the conventional focus on network capabilities, this thesis proposes a novel *user-network oriented* approach. This approach allows user terminals to perform control functions and to make use of service-specific information, such as the nature of calls and the nature of information within connections of the calls, in order to reduce call establishment delay and to minimise the bandwidth required by the connections.

With B-ISDNs being expected to support general multiparty multimedia calls, a call control scheme is required to process such calls, and to establish and to manage the association among the parties (users) and the connections associated with each call. To this end, a hierarchical call control structure is proposed in this thesis. The structure allows the control functions to be carried out either by the network or by the user terminals depending on the level of terminal intelligence. The details of the call control structure along with the required signalling protocol are described. Examples of both simple and complex call establishments are provided in order to illustrate the proposed call control.

Within each connection of a call, *cell loss priority (CLP)* can be used to identify cells having different loss probability requirements. Users can use CLP for tagging cells containing less essential information. Furthermore, we propose the possibility for users to police their traffic appropriately and to selectively tag any excess cells, considered expendable or else protected by end-to-end error recovery schemes, as being low priority; this avoids indiscriminate cell losses that would be caused by network usage of CLP.

Not all connections will allow tagging of cells. Therefore, based on the existence or absence of *pretagged* or low priority cells in a connection, we can distinguish two classes of connections, namely connections without pretagged cells (*the class of pure connections*) and connections with pretagged cells (*the class of mixed connections*). The traffic management framework for pure connections is very well established, but this is not the case for mixed connections. Therefore it is of the interest in this thesis to present a traffic management framework for mixed connections, which includes connection admission control, usage parameter control, as well as buffer management and scheduling policy.

Connection admission control algorithms, based upon a virtual bandwidth concept, involve a search for an equivalent bandwidth required by a connection. With mixed connections, there are two different QoS requirements. Existing methods which satisfy both requirements are either specific to some services, or time consuming, due to the required separate search for sufficient bandwidth to satisfy each QoS requirement. Considering these drawbacks, we propose two bandwidth allocation schemes in a homogeneous traffic environment and one scheme in a heterogeneous traffic environment. The methods require a single search for bandwidth and fully exploit the statistical dependency between the high and low priority traffic, and, as the result, allocate smaller bandwidth than previously proposed methods.

After a connection has been admitted, a usage parameter control or policing algorithm is required, to monitor and to control the traffic within the connection during the information transfer phase, in order to ensure that the negotiated traffic parameters are not exceeded. The leaky bucket algorithm is the basis for the most popular policing schemes. In order to make sense of the large number of leaky bucket schemes, two classifications are proposed. Most of the schemes can not take into account the presence of pretagged cells in mixed connections. Therefore, in order to overcome this drawback, we propose four modifications to the original leaky bucket scheme. Comparative studies based on both analytical and simulation techniques show that the newly proposed leaky bucket schemes can offer better quality of service for high priority traffic than earlier proposed leaky bucket schemes.

The convergence within a network of high priority cells in pure connections and both high and low priority cells in mixed connections can lead to the interference between the cells in the sense that overload of low priority cells in a mixed connection can degrade not only the QoS of other mixed connections but also the QoS of pure connections. Therefore it is necessary for the network to implement a buffer management scheme to minimise such interference, while simultaneously trying to maximise the utilisation of network resources. A new buffer policy, called dual queues with limited cyclic service (DQCS), is developed and shown to achieve both objectives by both discrete-time performance analysis and simulation.

The thesis concludes with proposals of some novel schemes for further study in order to develop an overall call control and traffic management framework for B-ISDNs.

ACKNOWLEDGEMENTS

First of all, I wish to express my most sincere gratitude to my supervisors, Associate Professor Harsha R. Sirisena and Dr. Krzysztof Pawlikowski for their guidance and encouragement throughout the course of my research. Their many suggestions have helped to improve the presentation of this thesis.

Special thanks are due to Mr. William K. Kennedy for his enthusiasm and encouragement, especially during the sabbatical leave of my supervisors, and also due to Mr. Ross Wallace for his spiritual support and friendship.

I wish to thank many staff members and postgraduate students in the Department of Electrical and Electronic Engineering, University of Canterbury. In particular, I would like to express my thanks to Associate Professor David B. Watson for his advice and help during my undergraduate years, Dr. Philip J. Bones for arranging the opportunities for me to supervise the Control laboratories, Mr. Ron Battersby for his patience and help while I was supervising the laboratories, Mr. Dave van Leeuwen for the technical help with the SUN workstations and Mr. Mike Waterman for some valuable discussions.

The willing assistance provided by all of the Engineering Library and the Physical Science Library staff is gratefully acknowledged, and the fellowship of brothers and sisters in the International Bible Study Group is also highly appreciated.

I acknowledge financial support from Telecom New Zealand Limited and I am grateful for the travel award I received from the IEEE Communication Society.

Last, but by no means least, I give my deepest thanks to the members of my family for their love, support and encouragement; especially my parents for providing me an opportunity to study in New Zealand and for supporting me spiritually and financially, and my wife, Agnes, for her devoted encouragement and understanding.

CONTENTS

GLOSSARY

Mathematical Notation

$\triangleq$	defined as
$\approx$	approximately equal to
$\rightarrow$	approaches
$\mathbf{A}$	matrix or vector
$A(i)$	ith entry of vector $\mathbf{A}$
$A(i, j)$	entry of matrix $\mathbf{A}$ in the ith row and jth column
$\mathbf{1}$	matrix or vector of ones
Prob $[X]$	probability of event X
$x(k)$	probability mass function of X
$E[x]$	expected value of random variable X
$\lfloor x \rfloor$	the smallest integer greater than x
$\lceil x \rceil$	the largest integer less than x
$\lvert x \rvert$	the absolute value of x

Acronyms

The following acronyms are used in general discussions of ATM networks. Most of the abbreviations are standard, but they are included for completeness.

AAL	ATM adaptation layer
ATM	asynchronous transfer mode
B-ISDN	broadband integrated services digital network
BMS	buffer management and scheduling
CAC	connection admission control
CBR	constant bit rate
CCITT	International Telegraph and Telephone Consultative Committee
CCT	cross-connect table
CDV	cell delay variation
CLP	cell loss priority
CPE	customer premises equipment

CPN	customer premises network
FIFO	first-in-first-out
GFC	generic flow control
IN	intelligent network
ISDN	integrated services digital network
kbps	kilobits per second
LIFO	last-in-first-out
Mbps	megabits per second
MIN	multistage interconnection network
ms	miliseconds
NNI	network-network interface
OSI	open system interconnections
PABX	private automatic branch exchanges
PDU	protocol data unit
PRM	protocol reference model
QoS	quality of service
SCP	service control point
SE	switching element
SVC	signalling virtual channel
STM	synchronous transfer mode
UNI	user-network interface
UPC	usage parameter control
VBR	variable bit rate
VC	virtual channel
VCI	virtual channel identifier
VP	virtual path
VPI	virtual path identifier

The following acronyms are used in discussing control mechanisms. They are defined at the beginning of each section in which they are used.

		Typical Reference
AC	adaptive control	p. 157
AT	aggregate traffic	p. 77
BC	bearer control	p. 34
BDLBM	buffered dual leaky bucket with marking	p. 92
BDLBMP	buffered dual leaky bucket with marking and priority	p. 98
BDLBP	buffered dual leaky bucket with priority	p. 96
BDLBP-DT	buffered dual leaky bucket with priority and dynamic threshold	p. 96
BLB	buffered leaky bucket	p. 89
BLBM	buffered leaky bucket with marking	p. 91
BLBP	buffered leaky bucket with priority	p. 94

CBS	complete buffer sharing	p. 125
CC	call control	p. 36
CCBM	combined cell-level and burst-level multiplexer	p. 155
CCBP	combined cell-level and burst-level policer	p. 157
CRR	class related rule	p. 76
DLBM	dual leaky bucket with marking	p. 90
DLBMP	dual leaky bucket with marking and priority	p. 97
DLBP	dual leaky bucket with priority	p. 93
DQCS	dual queues with limited cyclic service	p. 134
ESPP	extended simulated protective policy	p. 144
FRP	fast reservation protocol	p. 155
FRP/DT	fast reservation protocol with delayed transmission	p. 155
FRP/IT	fast reservation protocol with immediate transmission	p. 155
HDQCS	hierarchical dual queues with limited cyclic service	p. 158
ID	identifier	p. 30
IPP	interrupted Poisson process	p. 52
LBM	leaky bucket with marking	p. 89
LPO	limited push-out	p. 142
MMPP	Markov modulated Poisson process	p. 53
NP	network policer	p. 157
OLB	ordinary leaky bucket	p. 87
PBS	partial buffer sharing	p. 126
PC	party control	p. 35
PO	push-out	p. 127
PT	pointer	p. 30
RS	route separation	p. 129
SPP	simulated protective policy	p. 143
TPO	threshold push-out	p. 142
UC	user control	p. 36
UNP	user-network policer	p. 156
UP	user policer	p. 157

PREFACE

Over the last decade, we have seen an increasing interest from both telecommunication and computer industries in telecommunication networks which can offer integrated services for a wide spectrum of applications. This is exemplified by standardisation and deployment of *integrated services digital network (ISDN)* to support data services in traditionally voice networks and the upgrading of *fiber distributed data interface (FDDI)* to FDDI-II to allow integration of voice services with transmissions of computer data. With the advent of *asynchronous transfer mode (ATM)* technology, the path towards full service integration is opened widely. The acceptance of ATM technology for future *broadband-ISDN (B-ISDN)* by the International Consultative Committee for Telephone and Telegraph (CCITT) in 1988 has spearheaded a tremendous research effort worldwide on various aspects of ATM networks, such as congestion control, network architecture, switching, traffic modelling, communication protocols and signalling.

Around this time, the author was introduced to the opportunities for participating in research this challenging area by Associate Professor Harsha Sirisena. The author enrolled in the Ph.D. program in November 1990 and commences his research work under the guidance of Associate Professor Sirisena and Dr. K. Pawlikowski. The following narrative describes how the work reported in this thesis came about.

Initially the focus of my research was on traffic control in ATM networks, especially those related to the use of *cell loss priority (CLP)*. Although CLP allows the network to maximise resource utilisation, it also can cause some adverse effects. The initial study, presented by the author at SICON'91, showed that overloading by low priority cells can degrade the performance of high priority cells in existing buffer management schemes. The desire to reduce this interference, yet at the same time maximising the network utilisation, led us to the proposal of *dual queues with limited cyclic service (DQCS)* scheme in the paper. The scheme has been further developed and is described in Chapter 5 of this thesis.

The use of CLP for marking excess cells by a network policing scheme is questionable for many reasons. The main concern is that the network does not have knowledge about the significance of the information contained in a cell and its relationship to other cells in a connection as the control is only performed at the ATM layer. This can lead to cells containing the most essential information of video traffic or a cell from a long data message being marked and discarded within the network, causing degradation of video quality or cascading of cell losses experienced by users of data services. Furthermore, there is no distinction possible between the cells marked by the network as excessive cells and the cells tagged by users as containing less essential information, hence untagged cells that have been marked will be treated by the network in the same way as tagged cells. This is obviously undesirable since obviously users prefer the network to drop the tagged cells before the marked cells. In addition to these technical disadvantages, users would also be reluctant to be charged for cells marked and dropped within the networks.

An initial idea of providing an additional bit for differentiating marked cells from tagged cells was rejected because it increases the complexity of both policing and buffer management schemes. An alternative later adopted was for users to police their cells properly and to *selectively mark or tag* their cells. The cells can either contain less essential information and so expendable under congestion conditions, or protected by end-to-end error recovery schemes. Through this alternative, users are fully aware as to which cells are more likely to be dropped and so does the network, which can charge different rates for the tagged and the untagged cells. If the users have policed their traffic properly, then the network simply needs to monitor the traffic to ensure the correctness of the user policing and to simply discard any excess cells without any further marking. This general policing framework, callled *user-network policer (UNP)*, was presented at the INFOCOM'93.

In this framework, the network policer should be able to differentiate the tagged cells from untagged cells. Existing policing schemes, however, can not do so. Therefore modifications of existing leaky bucket schemes have been considered. One such modified scheme, called a *buffered leaky bucket with priority (BLBP)*, has been presented at the ABSSS'93. Four further modifications are presented in Chapter 4.

The suggested CLP applications by users have lead to the possibility of two classes of connections, namely connections without pretagged cells and connection with pretagged cells. We refer to them as *pure* and *mixed* connections, respectively. The bandwidth allocation, as part of connection admission control, for pure connection has been widely studied, but this is not the case for mixed connections. This motivated us to investigate these issues, which has resulted in the development of three bandwidth allocation methods for this class of connections as reported in Chapter 3. A paper reporting this work will be presented at the ATNAC'94.

When researching the connection admission control, the author realised that the term call admission control is often misused when the intended reference is connection admission control. In traditional networks, where a call consists of a single connection, the two terms are synonymous. However, as services in the future B-ISDNs will include multiple parties and multiple connections, there are clear differences between the two concepts. This leads the author to initiate research in the call control aspects. The call control protocols, which have been previously proposed, are normally defined primarily for the most complex calls, such as conference calls. Additional overhead is experienced by less complex calls. Furthermore, the dependency of users on the network to provide the necessary signalling capability limits user services. The desire to minimise this overhead, while providing more flexibility for users to support various services through their own customisation of a service has resulted in a hierarchical call control structure as presented by the author at ABSSS'92 and extended in a presentation at GLOBECOM'93. The call control is reported here in Chapter 2.

The title of this thesis summarises the two aspects of this research. The phrase *user-network oriented* was chosen to denote the approach developed here, which is focused on introducing user terminals' involvement in the *call control* and *traffic management* process, unlike in the conventional approach where all the control functionalities reside in the network.

In carrying out the research, both simulation and analytical techniques have been used. All analysis has been done by using MATLAB version 4.0 running on SUNSPARC computers and through programming in C language. In choosing tools for performing simulation studies, two criteria were used by the author, namely: reusability of simulation models due to a number of comparison studies among various schemes that had to be done, and proper statistical analysis in order to ensure the correctness of results for comparison. The simulation packages the author came across have met either one or the other but not both criteria. The desire to provide a package which can statisfy both criteria has led to the development of DESC++, a discrete event simulation package in C++, described in the Appendix. A paper describing the package has been submitted for publication. Using DESC++, we were able to obtain all simulation results with the relative precision below 0.05 at a 95% confidence level.

Organisation of this Thesis

This thesis comprises six chapters. Four of the chapters, viz. Chapters 2 to 5, contain the original works of the author, while Chapter 1 contains a survey of the state of the art of call control and traffic management and Chapter 6 contains some novel schemes proposed by the author for further study in order to develop an overall call control and traffic management framework.

Chapter 1 introduces the B-ISDN networking technology by reviewing its history and some of its key characteristics, which includes the services to be supported, the underlying ATM technology, the B-ISDN Protocol Reference Model (PRM) and the ATM Adaptation Layer. It also provides a general overview of the call control and traffic management mechanisms that have been proposed in the literature. For details of these mechanisms, readers are referred to appropriate papers. The chapter concludes with a statement of the aims and the major contributions of this thesis.

Chapter 2 presents a new model for B-ISDN calls and a new signalling architecture and call control structure to allow B-ISDNs to support generic multiparty multimedia calls. The chapter also introduces additional multicast switch capabilities for supporting conference services. Both analytical and simulation results are provided to illustrate possible gains from the control structure for both simple and complex calls.

Chapter 3 considers bandwidth allocation for connections with pretagged traffic (mixed connections). Individual sources are modeled by *interrupted Poisson process (IPP)* and their superposition is approximated by a *Markov modulated Poisson process (MMPP)*. Three recently proposed methods for matching the characteristic of the superposition to the parameters of MMPP model are compared. The analysis of a multiplexer fed by the resulting MMPP is carried out on a discrete-time basis. Based on the analysis, six methods (Methods V and VI are new) for allocating bandwidth in a homogeneous environment and two methods (AT method is new) in a heterogeneous environement are compared.

Chapter 4 introduces ways of classifying existing leaky buckets and describes twelve schemes (DLBP, BDLBP, BDLBP-DT, BDLBMP schemes are new) out of all possible combinations of discarding, buffering, marking and priority mechanisms used in a leaky bucket scheme. The performances of the schemes in policing an individual mixed connection (e.g. a bursty video source) is analysed on a discrete-time basis. In addition to studying the leaky buckets in isolation, performance of a multiplexer fed by a number of policed sources and end-to-end quality of service experienced by a source are also compared using simulation.

Chapter 5 describes four classical buffer management schemes to be located at the convergence points of traffic of pure and mixed connections within the network. Discrete-time analysis of each scheme is presented and a general protection criterion for comparing the schemes is proposed. A new scheme, DQCS, is developed and analysed on a discrete-time basis for evaluating its performance. The scheme is also compared with four recently proposed schemes using simulation.

Conclusions are drawn in Chapter 6 and some suggestions for future research are made. In the area of call control, both intramedia and intermedia synchronisation play curcial roles in providing an overall call control framework. Existing works are reviewed and a possible scheme to facilitate both types of synchronisation is proposed for further study. In the area of traffic management, there is a need for a control scheme to bridge the cell-level control often suggested for voice and video services and the burst-level control suggested for data services. Therefore a combined cell-level and burst-level control scheme is proposed for further study and based on this control scheme and other schemes proposed in this thesis, an overall traffic management configuration for B-ISDNs is presented for further research.

List of Publications

The following publications have been prepared during the course of this research.

HARTANTO, V.F., SIRISENA, H.R., PAWLIKOWSKI, K. and KENNEDY, W.K. (1991), 'Performance Study of Dual Queues with Limited Cyclic Service in ATM Switching', In *Proc. of Singapore Int. Conf. on Networks (SICON'91)*, Singapore, September, pp. 253–258.

HARTANTO, V.F., SIRISENA, H.R., PAWLIKOWSKI, K. and KENNEDY, W.K. (1992), 'Dissecting Call Establishment Procedures in ATM Networks', In *Proc. of Australian Broadband Switching and Services Symposium (ABSSS'92)*, Melbourne, Australia, July, pp. 605–612.

HARTANTO, V.F. (1993), 'DESC++: Discrete Event Simulation Library using C++ Language', *Technical report*, University of Canterbury, New Zealand, January.

HARTANTO, V.F. and SIRISENA, H.R. (1993), 'User-Network Policer: A New Approach for ATM Congestion Control', In *Proc. of IEEE INFOCOM Conference*, San Francisco, CA, March, pp. 376–383.

HARTANTO, V.F. and SIRISENA, H.R. (1993), 'Message Loss Study of Leaky Bucket Algorithms', In *Proc. of Australian Broadband Switching and Services Symposium (ABSSS'93)*, Wollongong, Australia, July, pp. 9–17.

WATERMAN, M., HARTANTO, V.F. and SIRISENA, H.R. (1993), 'Voice Traffic Control in B-ISDNs', In *Proc. of Australian Broadband Switching and Services Symposium (ABSSS'93)*, Wollongong, Australia, July, pp. 38–45.

HARTANTO, V.F. and SIRISENA, H.R. (1993), 'A Framework for B-ISDN CPE and Signalling', In *Proc. of IEEE GLOBECOM Conference*, Houston, TX, November, pp. 1511–1515.

HARTANTO, V.F., SIRISENA, H.R. and PAWLIKOWSKI, K. (1994), 'Comparative Study of Fast Reservation Protocols and Adaptive Input-Rate Flow Controls, *accepted for presentation at the Australian Telecommunication Networks and Applications Conference (ATNAC'94)*, Melbourne, Australia, December.

HARTANTO, V.F., SIRISENA, H.R. and PAWLIKOWSKI, K. (1994), 'Equivalent Bandwith for Connections with Pretagged Traffic, *accepted for presentation at the Australian Telecommunication Networks and Applications Conference (ATNAC'94)*, Melbourne, Australia, December.

HARTANTO, V.F., PAWLIKOWSKI, K., SIRISENA, H.R. and KREUTZER, W., 'DESC++: An Object-Oriented Tool for Simulating Telecommunication Networks', *submitted for publication in Computers and Electrical Engineering*.

Chapter 1

INTRODUCTION

Traditionally, networks have been built for specific applications. This causes a network for a particular application to be less suited to another, since it operates under different constraints. For example, the public telephone network, designed for carrying voice communications by reserving dedicated network resources to a call, is less efficient when being used to carry bursty data. On the other hand, X.25 packet network, which is specifically implemented for data communications, is unsuitable for carrying voice calls due to its large and variable transfer delays, and non-guaranteed throughput.

With the rapid growth in demand for new applications in the information age, the current practice of responding to diverse needs by developing application-specific communication networks, or by deploying several of them in parallel, is obviously unworkable. Firstly, it is inefficient and expensive to maintain several networks. Secondly, this solution is unlikely to satisfy long term needs [Turner, 1986]. For these reasons, the telecommunication industry has been interested in future public networks based upon a form of switching and multiplexing (referred to by CCITT or TSS[1] as transfer mode) that has the potential for unifying the multitude of separate networks in existence today as well as catering for any new application demands in the future. This network is expected to provide a generally useful set of communication capabilities in an application-independent fashion as a foundation to build a large class of new future applications. In comparison with separate dedicated networks, service and network integration has major advantages in economic planning, development, implementation, operation, and maintenance [Armbruster and Schaffer, 1986].

The effort towards building an integrated network came to fruition with the deployment of *integrated services digital networks (ISDNs)* in recent years [ISDN, 1991, 1992]. The ISDN provides users with digital line interfaces, viz. basic rate interface (BRI), which comprises two 64 kbps circuit-switched or B-channels for voice and bulk data transfer, and a 16 kbps packet-switched or D-channel for data and control information, and primary rate interface (PRI), which comprises 23 B-channels and one D-channel (at 64 kbps) in North America and Japan or 30 B-channels and one D-channel (at 64 kbps) in Europe. ISDN is, however, not really a fully integrated network, since it involves integration only at the loop and even here separate channels are prescribed. The communication required to support this interface must provide separate switching mechanisms for circuit switching and packet switching channels. It is further limited by low speed (not suitable for high and variable bit rates services) and lack of flexibility.

[1]TSS stands for Telecommunication Standardisation Sector which is the new name of CCITT since March 1993.

In recent years, the increasing demand of speed beyond the ISDN rates has provided the impetus for *broadband-ISDNs (B-ISDNs)* [Day, 1991]. This is made possible by advances in telecommunication technologies which offer abundance of bandwidth from fibre optic technology combined with advanced switching techniques and intelligence to manage them within the networks. The demand for broadband networks comes from a number of applications, such as the existence of today's high bandwidth customer premises networks (i.e. local area network (LAN)) requiring long distance broadband interconnections [Lindisky, 1990; Amin-Salehi *et al.*, 1990], and the emerging field of radiology in which sharing medical images among hospitals, physicians, and patients requires large bandwidths due to the enormous data files [Akselsen *et al.*, 1993].

An overview of B-ISDN services and their characteristics is provided in Section 1.1. In order to cater for such wide variations of service characteristics, CCITT [1990a] has chosen *Asynchronous Transfer Mode* (ATM) as the most suitable and unified information transport technique for B-ISDNs. ATM allows information to be transferred across the networks in a service-independent manner. The details of ATM technology are presented in Section 1.2. The ATM layer must be accompanied by a set of higher layer protocols and network controls in order to support services outlined in Section 1.1. These higher protocols are defined through the B-ISDN protocol reference model (PRM). The PRM is described in Section 1.3, followed by the details of ATM adaptation layer (AAL). The network control aspects include signalling and call control, traffic management, and operations, administration, and maintenance (OAM). An overview of call control and traffic management mechanisms that have been proposed so far is given in Sections 1.4 and 1.5, respectively. The concluding Section 1.6 summarises the aims and the major contributions of this thesis.

1.1 B-ISDN Services

In the following discussion, a *service component* refers to the type of media used in a service and a *connection configuration* refers to the way connections are set up between communicating users.

1.1.1 Service Component and Connection Configurations

In addition to the services comprising a single component and requiring a single connection between two parties (users), services in the target B-ISDNs may comprise more than one service component. Each component transports a specific type of information and can be specified by widely varying characteristics due to various coding and compression techniques used. They are called *multimedia* services, such as video telephony services. Also as in other communication networks, B-ISDN services may involve more than two parties. These general services are called *multiparty multimedia* services. A number of possible services with their respective service components is shown in Table 1.1 [De Prycker, 1991], while the required connection configurations for the services is shown in Figure 1.1.

The following service components can be distinguished

- *Audio* components, transporting the speech or audio signal with various audio quality and bit rate requirements, from the classical PCM (64 kbps) to the high quality sound generated by a CD player.

B-ISDN services	Service component: Audio	Data	Video	Connection configuration†
Voice telephony	✓			(a)
High speed data		✓		(a)
Video database	✓	✓	✓	(a) and (b)
Video conference	✓	✓	✓	(c) and (d)
Broadcast video	✓	✓	✓	(c) and (d)

†refers to Figure 1.1.

Table 1.1 B-ISDN service components and connection configurations.

- *Data* components, including data information, both graphics and text, and LAN-to-LAN interconnections. In the case of video services (TV distribution), the data components can include subtitling and transferring information related to the type of programme.

- *Video* components, transferring the standard video image, which can be video signals with an average bit rate value around 15 to 20 Mbps. A lower bit rate (e.g. around 1 to 5 Mbps) may be used in video conference services.

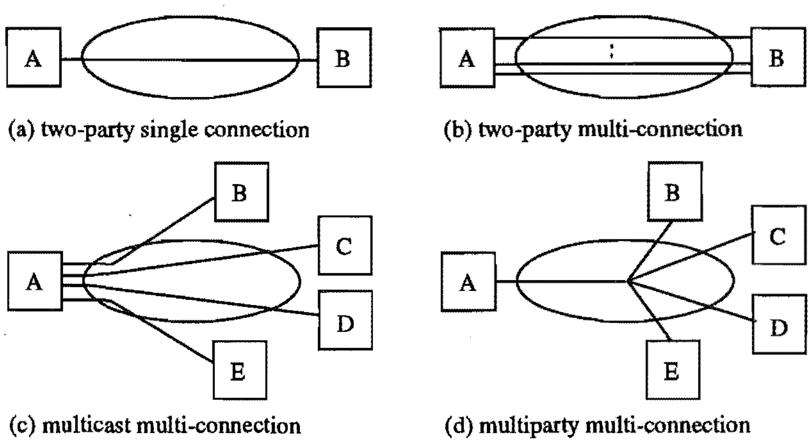


Figure 1.1 B-ISDN connection configurations.

The connection configurations required for carrying the services (see Figure 1.1) can be classified as

- *Point-to-point (pp)* configurations, involving either a single point-to-point connection (Figure 1.1(a)) or multiple point-to-point connections (Figure 1.1(b) and (c)) between two parties. The connection used in the communication can either be unidirectional or bidirectional. In the case of bidirectional connection, it can be symmetric or asymmetric.

- *Multipoint (mp)* configurations, involving either a point-to-multipoint connection from a party to multiple parties or a multipoint-to-multipoint connection among all parties (Figure 1.1(d)).

1.1.2 Service Classification

With the vast range of possible B-ISDN services as described in the previous section, CCITT [1990b] has classified services into interactive and distributive services (refer to Figure 1.2).

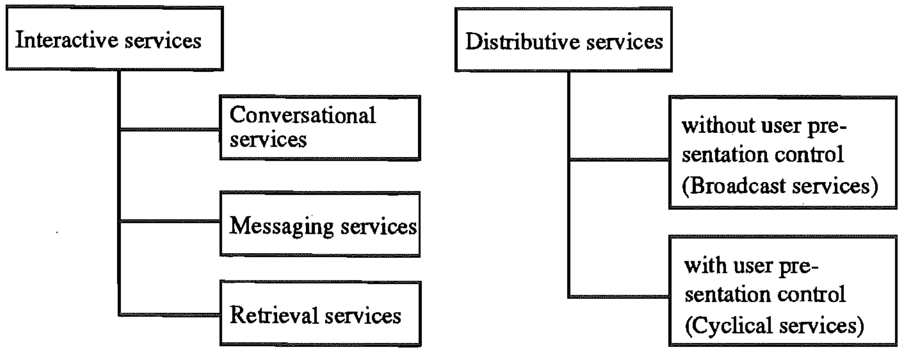


Figure 1.2 B-ISDN service classification.

Interactive services require point-to-point connections between customers which can be symmetrical for *conversational* services, such as videotelephony, or asymmetrical for *messaging* and *retrieval* services, such as document transfer and retrieval. *Distributive* services require point-to-multipoint connections from a service provider (e.g. video studio) to authorised customers. These include *broadcast* services, for which the user has no control over the presentation of the information, and *cyclical* services, which allow users some measure of presentation control. A variety of potential interactive and distributive broadband services are listed in [Handel, 1989; Stallings 1989; De Prycker, 1991].

In addition to this classification, there are various other ways of classifying B-ISDN services. One common way is to classify the services based on their characteristics, such as narrowband and broadband signals; bursty and continuous traffic; connection-oriented and connectionless; point-to-point and complex communications; switched and non-switched; real-time and non-real-time [Minzer, 1989]. A service may belong to a number of classes, e.g. a service could be interactive, narrowband and non-real-time such as voice messaging service and each service class may be divided further into subclasses. For example, broadband services can be subdivided by the type of connection required as video communication, image retrieval, LAN interconnection and multimedia communication [Roberts, 1991]. Another possible way is to classify them according to whether they provide access to human beings or to computers [Patterson and Egido, 1990].

1.2 ATM Technology

ATM is a connection-oriented protocol that supports both connection-oriented and connectionless services. It operates on the principle of packet switching, which allows establishment of several virtual connections of totally different bit rates via a single access and uses a uniform network switching equipment. An ATM transport network comprises both access and intermediate nodes, which are interconnected by fiber optic links. Users gain access into the network via a standard *user-network interface (UNI)* as shown in Figure 1.3. Their information is transported through the network in fixed length packets or cells. In the following sections, we will review the ATM connections, cell structure, and switching architectures.

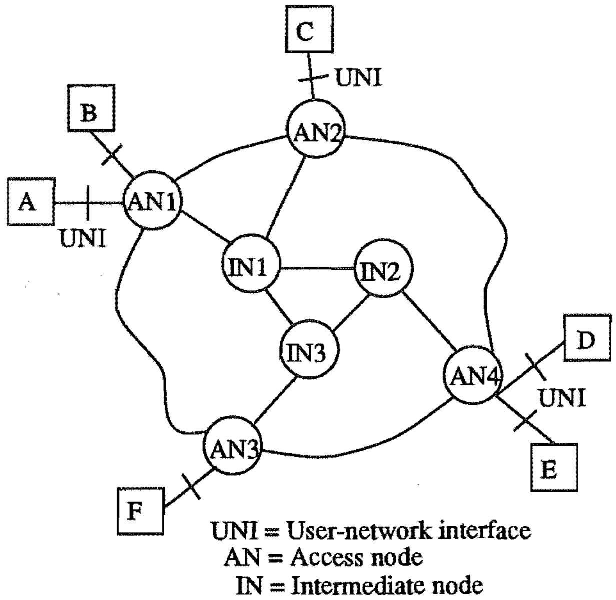


Figure 1.3 Example of ATM transport network.

1.2.1 ATM Connections

A physical link in an ATM network may be segregated into several virtual paths and each virtual path may carry several virtual channels as shown in Figure 1.4.

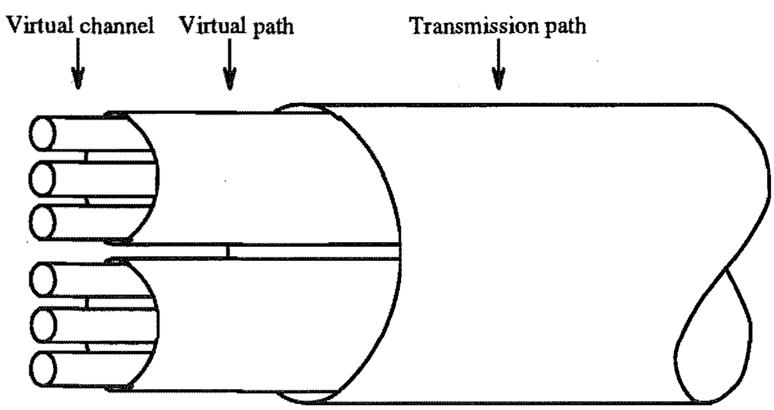


Figure 1.4 Relationship between virtual channel, virtual path and ATM link.

A *virtual channel (VC)* is the basic unidirectional connection for transporting ATM cells. It is established on demand and is characterised by a *virtual channel identifier (VCI)*. Within a virtual channel, delivery of ATM cells is always guaranteed to be in sequence.

Virtual channels that share a common route through the network are bundled together into a *virtual path (VP)* in order to overcome the overheads of a large number of virtual channels being switched individually [Ohta *et al.*, 1988; Burgin, 1989]. Unlike virtual channels, a virtual path is established semi-permanently between endpoints of the virtual channels and is identifed by a *virtual path identifier (VPI)*. An example of the virtual path usage (adopted from [Sutherland *et al.*, 1988]) is described in Figure 1.5.

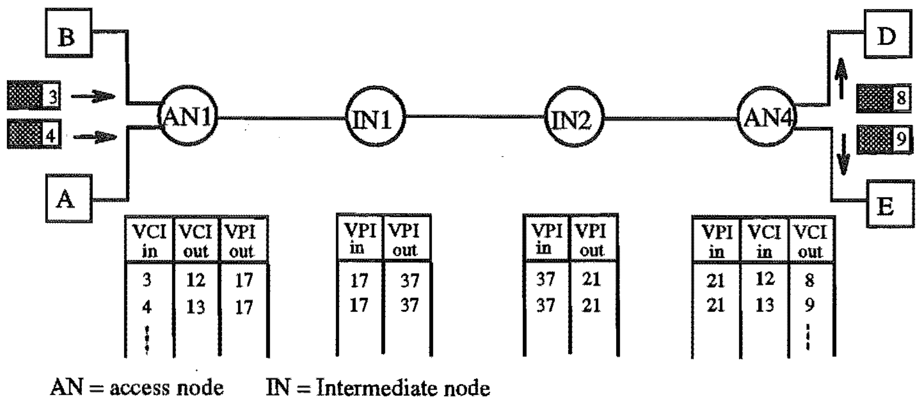


Figure 1.5 Example of VPI use in an ATM network.

The diagram shows two terminals, A and B, which set up virtual channels to terminals D and E. The virtual channels are identified by VCI 3 and 4, respectively. At the access node AN1, cells from both virtual channels are multiplexed onto a virtual path identified by VPI 17. The virtual path links nodes AN1 and IN1. At the intermediate node IN1, the cells are switched onto another virtual path connecting IN1 and IN2, which is identified by VPI 37. The switch at node IN1 translates the VPI of the cells only, while leaves their VCI values intact. When the cells arrive at the remote node AN4, their incoming VPI is stripped off and their incoming VCIs are mapped to the corresponding outgoing VCIs and finally the cells are routed to the destination terminals.

1.2.2 ATM Cell Structure

Within an ATM connection, information is carried in short and fixed length packets, called cells. Each cell consists of a 5 octet header field and a 48 octet information field. The cell header function at the UNI differs from that at the *network-network interface (NNI)* in the use of bit 5-8 of octet 1 as shown in Figure 1.6.

The field descriptions are as follows [CCITT, 1992a].

- *Generic flow control (GFC)* can be used at the UNI to assure proper access of various terminals to a *customer premises network (CPN)* in order to alleviate short-term overload conditions. The format of this field is dependent on the configuration of the CPN [Zukerman *et al.*, 1991].

- A concatenation of *virtual channel identifier (VCI)* and *virtual path identifier (VPI)* forms a complete label, but not an explicit address, for routing a cell through an ATM connection.

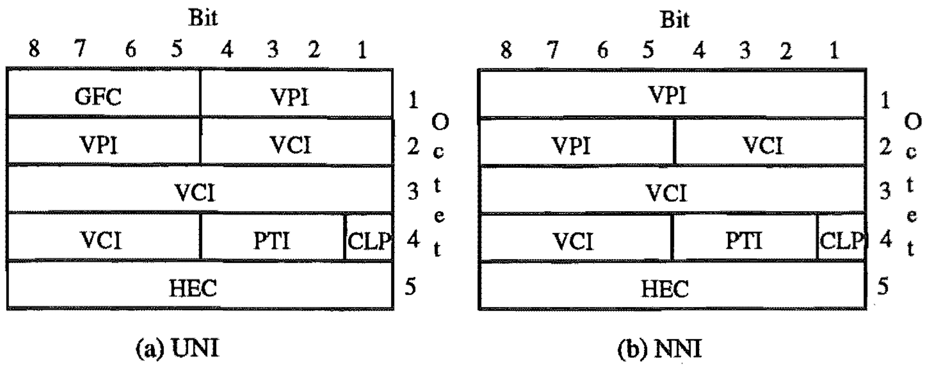


Figure 1.6 ATM cell header at the UNI and the NNI.

- The *payload type identifier (PTI)* makes it possible to distinguish between user information and network information (e.g. maintenance information) within an ATM connection.

- By means of the *cell loss priority (CLP)* users may indicate to the network which cells within an ATM connection contain less significant information. The CLP can also be used by the networks to mark cells which violate the negotiated traffic parameters. The cells with $CLP = 1$ will be discarded first under congestion conditions.

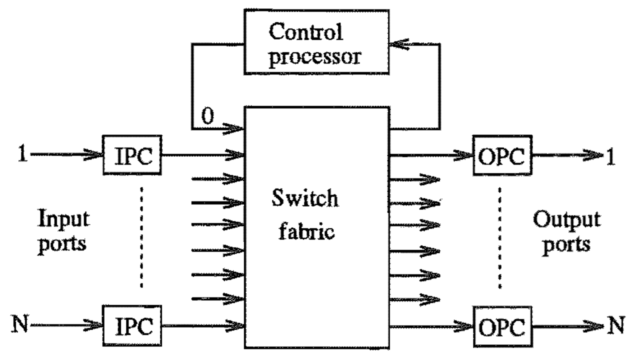
- *Header error control (HEC)* allows multiple bit error detection and single bit error correction in order to minimise misdelivery of cells in the ATM layer. It is also used for cell delineation. The details of cell delineation and header error control algorithm are described in [CCITT, 1990f].

The information field of an ATM cell may consist of headers as well as protocol data units from layers above ATM layer. The field is transported transparently within the network by the ATM layer without any processing, e.g. error control, being performed on it.

1.2.3 ATM Switching Architecture

Each node in an ATM network contains one or more switches. The switches route each ATM cell to the desired destination according to the routing label (VPI/VCI) at the header of the cell. The general structure of an ATM switch is illustrated in Figure 1.7. In an ATM switch, all of the per cell processing functions are performed in hardware by the input port controllers (IPCs), the switch fabric, and the output port controllers (OPCs).

The switch fabric is the key difference between various switch topologies. A survey of switch fabrics can be found in [Ahmadi and Denzel, 1989; Tobagi, 1990; Newman, 1992]. It seems to be generally agreed that multistage interconnection network (MIN) of simple switching elements (SEs) with self-routing capability forms the most efficient switch fabric for broadband packet switches. Scores of different MINs have been reported to date, varying in their complexity and arrangements of the switching elements. For example, the Starlite switch [Huang and Knauer, 1984] and the Generalized Knockout switch [Eng and Karol, 1991] are two switch designs which are implemented by AT&T.



IPC = Input port controller; OPC = Output port controller

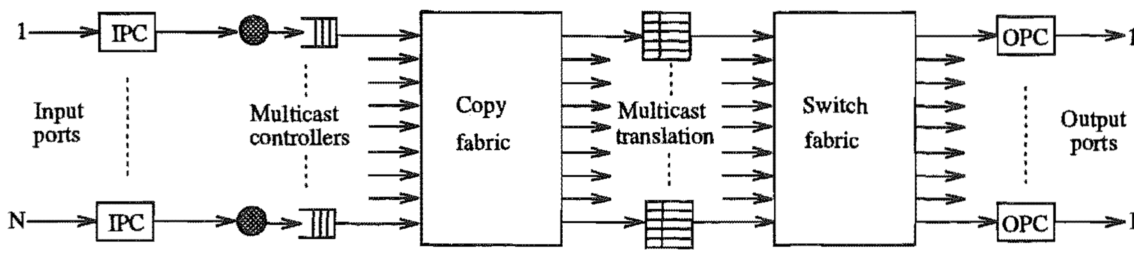
Figure 1.7 General structure of an ATM switch.

Besides the differences in the switch fabric, switch architectures can also be classified based on the queueing schemes which are used to resolve the conflicts when several cells are switched simultaneously from a number of input ports to the same output port. Three main queueing strategies are *input queueing*, *shared queueing*, and *output queueing*, see for example Pattavina [1993]. Input queue and output queue are located at the IPC and OPC, respectively, whereas the shared queue is available within the switch fabric to be shared by cells addressing any switch output. It is also possible for the switch to implement a combination of the approach. For example the Generalized Knockout switch implemented an output queueing, while the Starlite switch implemented a combination of shared and output queueing.

In addition to switch hardware, a control processor (CP) is required for higher-level functions such as connection establishment and release, bandwidth allocation, maintenance, and management [Newman, 1992]. During connection establishment, the CP configures the switch hardware to route incoming cells to the appropriate outgoing channels by modifying the *cross connection table (CCT)* within the switch. The switch routes control cells (as distinguished by the header) to the CP via port 0 (see Figure 1.7).

The switch architectures described so far support only unicast operation. As ATM networks are also required to provide point-to-multipoint connections for some B-ISDN applications, such as entertainment video distribution, and video conference, it is necessary for an ATM switch to implement a copy function in order to copy cells from one incoming channel to any subset of its outgoing channels. The switch which performs a copy function is called *broadcast* or *multicast* switch. It has an architecture as illustrated in Figure 1.8 [Lee, 1988b; Turner, 1988; Newman and Doar, 1990].

A broadcast switch generally has a copy fabric and a set of multicast controllers prior to the copy fabric. The multicast controllers add a copy tag to each of the incoming cells. The copy tag defines the number of copies of the cells that are required. The copy fabric replicates the cells according to the copy tag. All cells that exit from the copy fabric are then routed by the switch fabric to the required output ports. A label translation operation is necessary between the copy fabric and the switch fabric. This translation information are preconfigured during connection setup.



IPC = Input port controller; OPC = Output port controller

Figure 1.8 Broadcast switch architecture.

1.3 B-ISDN Protocol Reference Model

B-ISDN architecture [CCITT, 1990d], depicted in Figure 1.9, has been defined using the same layered structure as used in the open systems interconnection (OSI) model.

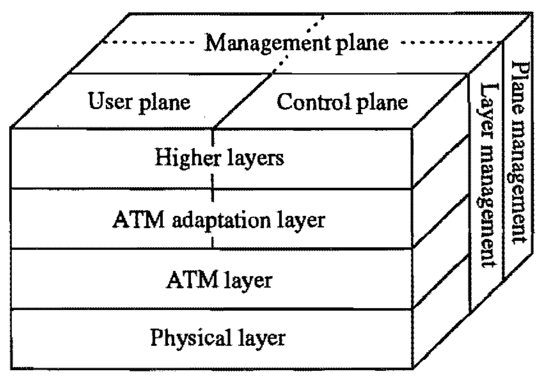


Figure 1.9 Generic B-ISDN protocol reference model.

In this architecture, information transfer capability common to all services is provided by the ATM layer. This layer uses the service of the *physical layer (PL)* in order to transport the cells. The physical layer is based on *synchronous digital hierarchy (SDH)* or *synchronous optical network (SONET)* transmission standards [Ballart, 1989]. Further information on the physical layer is contained in [CCITT, 1990f].

The service provided by the ATM layer for the upper layers is enhanced by the *ATM adaptation layer (AAL)*. The AAL performs some adaptation functions before delivering the cell payloads from the ATM layer to the next higher layers. Further descriptions of AAL will follow in the next section.

In addition to its layered structure, the B-ISDN protocol reference model also incorporates the concept of separated planes for the segregation of user, control and management functions. At higher layers, the user plane provides for user information flow transfer, with associate controls (e.g. flow control and recovery from errors), which is part of the overall traffic management strategies in B-ISDNs. On the other hand, the control plane performs the call control and connection control functions. It deals with the signalling necessary to set up, supervise and release calls or connections. Detailed discussions on the call control and traffic management will be given in Sections 1.4 and 1.5, respectively. For the details of management plane and its relations to the user and control plane, readers are referred to [CCITT, 1990d].

1.3.1 ATM Adaptation Layer

The *ATM adaptation layer (AAL)* is a service-dependent layer. It is responsible for mapping higher layer protocol data units (PDUs) into the information field of ATM cells at the transmitting ends and reassemblying those PDUs at the receiving ends. Additional information related to specific network service features, such as time-stamp field, sequence number, and coding type, are included in the AAL header.

AAL Type 1 AAL Type 2 AAL Type 3/4 AAL Type 5

	Class A	Class B	Class C	Class D
Timing betw. source & dest.	Required		Not required	
Bit rate	Constant	Variable		
Connection mode	Connection-oriented (CO)			Connection-less (CL)

Figure 1.10 AAL service classes and AAL types.

In order to cover the wide range of adaptation requirements in user and control planes, CCITT [1990e] has identified four service classes (see Figure 1.10) based on three parameters, namely timing relationship between source and destination (required/not required), bit rate (variable/constant) and connection mode (connectionless/connection-mode). Class A service supports connection-oriented services that require *constant bit rates (CBRs)*, and have specific end-to-end timing and delay requirements such as existing voice, future CBR video, and circuit emulation services. Class B supports connection-oriented services with *variable bit rates (VBRs)* and timing information between source and destination, e.g. VBR video services. Class C is VBR service intended for connection-oriented data applications, e.g. frame relay or X.25, while Class D provides VBR service for connectionless data applications, e.g. e-mail [McKinney and Gordon, 1994].

Initially, four types of AAL (Types 1–4) have been recommended to carry the different service classes [CCITT, 1990e]. But more recently, CCITT [1992b] has merged AAL Type 3 and AAL Type 4 into AAL Type 3/4 and introduced AAL Type 5. The service classes supported by these AAL are indicated by the arrows in Figure 1.10. In summary, AAL Type 1 is for Class A, AAL Type 2 is for Class B, AAL Type 3/4 are for Class C and D, while AAL Type 5 is for Class C. AAL Type 5 has a limited set of functions as compared to AAL Type 3/4 in supporting Class C services [Suzuki, 1994].

1.4 B-ISDN Call Control and Signalling

1.4.1 Call Establishment Procedures

The typical message flow in the process of establishing a call by the network is shown in Figure 1.11 [Chang *et al.*, 1992].

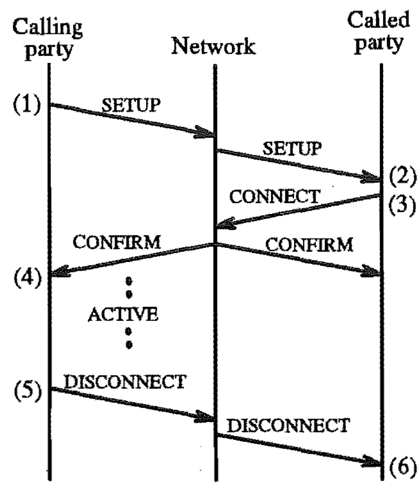


Figure 1.11 B-ISDN call establishment and release procedures.

Step 1. The calling party sends a SETUP message to the network, which at least contains the call description with one or more connection descriptions. Each connection description can include information about the connection, such as connection type, calling and called party addresses, bandwidth requirement, and coding scheme.

Upon receiving the SETUP message, the call control at the network parses the message into component transactions, derives the component services being requested, constructs a connectivity matrix of the parties participating in the call, and determines other service-specific attributes of each participant. In the process the call control may require additional service logic (e.g. supplementary service features to enhance a telephony service) and special resources (e.g. code conversion and information compression elements). These service logic and special resources are currently defined in relation to the *intelligent network (IN)* architecture [Duran and Visser, 1992] and are concentrated in a few dedicated nodes called *service control points (SCPs)* and *special resource functions (SRFs)*, respectively.

Step 2. The network sends SETUP messages to all called parties to notify them of the incoming call.

Step 3. Having offered an incoming call, a called party accepts or rejects the call by sending a CONNECT or a DENIED message back to the network.

Step 4. When replies from all parties have been received, the network decides whether to continue the call. If rejection from a party or a connection, which is vital to the call, has been received, then the network will send a DENIED message to the calling party and DISCONNECT messages to the other parties. Otherwise, the network sends CONFIRM messages to notify all parties that the call has been set up and is active.

Step 5. At the end of communications, a party terminates the call by sending a DISCONNECT message to the network.

Step 6. The network sends DISCONNECT messages to all other parties in the call.

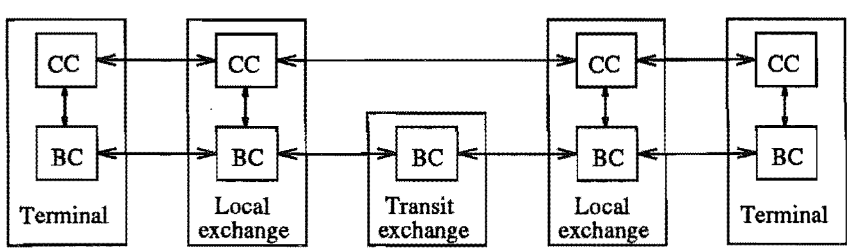
1.4.2 B-ISDN Signalling Architecture

As exemplified in the previous section, the establishment, control, and management of a call or connections in B-ISDNs require signalling, which is the exchange of information among the user and network entities. The existing signalling and call control protocols (e.g. Q.931 protocol for access signalling at the UNI) have a monolithic structure and allow users to specify just one connection description and control a single connection.

The need for supporting multiparty multimedia services requires the network to provide a signalling capability which allows users to specify more than one connection descriptions. Furthermore, more sophisticated call control and connection management functions within the network nodes are necessary to manage and control associations and connections between various parties in a call.

In satisfying these demands, CCITT [1990c] has standardised the use of ATM virtual channels for carrying signalling information at the UNI. These channels are referred to as *signalling virtual channels (SVCs).* The SVC can be of any bit rate, instead of fixed rate, as for the D-channel in narrowband-ISDN. The establishment of a SVC is performed on power-up of a terminal by a meta-signalling function operating at the ATM layer.

CCITT has further recommended the separation of the protocols to be used on the SVC into call and connection (bearer) components. The recommendation allows the call control to handle multiple connections, where each connection can be added, modified or removed during a call. The B-ISDN signalling architecture based on this standard is shown in Figure 1.12.



CC = Call control; BC = Bearer/connection control

Figure 1.12 B-ISDN signalling architecture.

The connection or bearer control (BC) module is required by all network nodes. The module has a basic connection control capability, such as connection establishment and signalling virtual channel allocation between a pair of exchanges. On the other hand, the call control (CC) module performs functions which are directly related to a call, such as call processing and coordination of individual BC modules associated with the call.

1.4.3 Design Approaches for B-ISDN Signalling

Recent work towards more sophisticated B-ISDN signalling has been based on two approaches, namely *revolutionary* approach and *evolutionary* approach.

So far, most work in this area has tended to use the revolutionary approach, where a new protocol is defined primarily for the most complex calls, such as conference calls. The protocol supports simpler calls as special cases of these complex calls. Some protocols based on this approach have been described in [Minzer, 1991; Bubenik *et al.*, 1991; Lim and Scott, 1991; Jeffrey and Sarma, 1992; Chang *et al.*, 1992]. Most of the protocols implement the separation of call and connection control to provide a unified protocol for supporting all services. One disadvantage of this approach is that it can result in excessive overhead and delay in setting up simple calls. The least overhead incurred is in determining the type of calls to be processed at the call control, whereas the delay is due to the queueing of call awaiting the longer processing time required by complex calls. A case of this will be illustrated in Section 2.5.

CCITT, on the other hand, prefers an evolutionary approach by recommending a stage delivery of B-ISDN signalling capability for various services [Rice and Spears, 1992]. In the early stage, an enhancement of current narrowband ISDN signalling protocols (e.g. Q.931) is recommended, followed by further extension to the standards to support multipoint connections.

In this thesis, the latter approach is adopted. Initially we define basic signalling capabilities for supporting point-to-point connections and extend these capabilities to support multipoint connections as a superset of the basic signalling protocol. In supporting complex calls, the choice of protocol sets to be used and the processing of the calls can be done either by the network or by the user terminals. The complete discussion of the protocols is given in Chapter 2.

1.5 B-ISDN Traffic Management

Each connection in ATM network is specified by its parameters, which include traffic descriptors and quality of services. Based on these parameters, the network allocates specific bandwidth to the connection. The allocated bandwidth in ATM connections has no explicit relation to any physical resources, such as dedicated slots within a frame or specific buffers. Due to this virtual allocation of bandwidth, ATM networks will generally behave as systems employing statistical multiplexing just like in the present packet switching networks.

As in any packet switching network, congestion is an inherent features of ATM networks due to the contention for network resources from several cells simultaneously. Cells that fail to seize the required resources will meet either variable delays or the risk of being discarded in case of buffer overflow. This situation can result in the degradation of service quality of the connection of which the cells belong to. To prevent such degradation, a congestion control scheme is required.

Numerous congestion control schemes have been proposed since the introduction of packet switching. A survey of such techniques can be found in [Gerla and Kleinrock, 1980; Pouzin, 1981; Schwartz, 1987]. Most of the techniques in used today are based on window mechanisms, which rely on the end-to-end exchange of control messages between sender and network nodes or receiver, in order to regulate the flow of packets into the network.

Although the mechanisms work well in low speed networks, they can no longer be relied upon to deliver efficient and stable control in high-speed ATM networks. This is principally due to the large ratio of propagation delay to transmission time inherent with the broadband speeds in ATM networks. Thus, the feedback from the network is usually too outdated to regulate the vast number of cells that may be in transit. This argues for congestion control mechanisms which do not rely too heavily on instantaneous network conditions. Instead, the mechanism is expected to use knowledge of the extrinsic parameters associated with a connection in order to regulate the traffic flow within the connection. Such a mechanism is generically termed traffic management.

The objective of ATM traffic management is to maximise network utilisation while maintaining specific quality of service for various B-ISDN services. In meeting this objective, the mechanims used should be simple enough for implementation in hardware, be adaptable and fair to diverse service requirements and diverse charateristics of ATM traffic, remain effective even if some of the assumptions are only partially valid, and be controllable [Eckberg *et al.*, 1989].

Currently, CCITT [1990c] has defined a family of traffic management mechanisms, which includes *connection admission control*, *usage parameter control*, *priority control*, and *reactive control*. Priority control often refers to the buffer management strategies implemented at the intermediate nodes to treat cells of various priority levels differently. However, in this thesis priority control can also be implemented in the usage parameter control at the access node, in order to treat high and low priority cells from users differently. Therefore, we prefer to use the term *buffer management and scheduling* instead of priority control in referring to the traffic management mechanisms at the intermediate nodes.

The relationship among the first three mechanisms is illustrated in Figure 1.13. The chapter number indicates the Chapter of this thesis where the mechanisms are investigated. Prior to describing each mechanism in details, we will first present discussions on connection parameters, viz. traffic characterisation and quality of services in the next section.

1.5.1 Connection Parameters

Traffic Characterisation

Traffic sources in an ATM network have a common multilayer structure with each layer being characterised by different time scales [Hui, 1988; Filipiak, 1989]. The most commonly used model is based on three levels of abstraction as shown in Figure 1.14. In the model, connections are composed of burst of cells and can stand for several minutes. The bursts are, in turn, composed of cells and last for a fraction of second. The cells take a few microseconds to be transmitted.

Each level of the model can be characterised by various parameters. Some of the parameters at the burst and cell levels, such as the burst duration and the inverse of the cell interarrival time

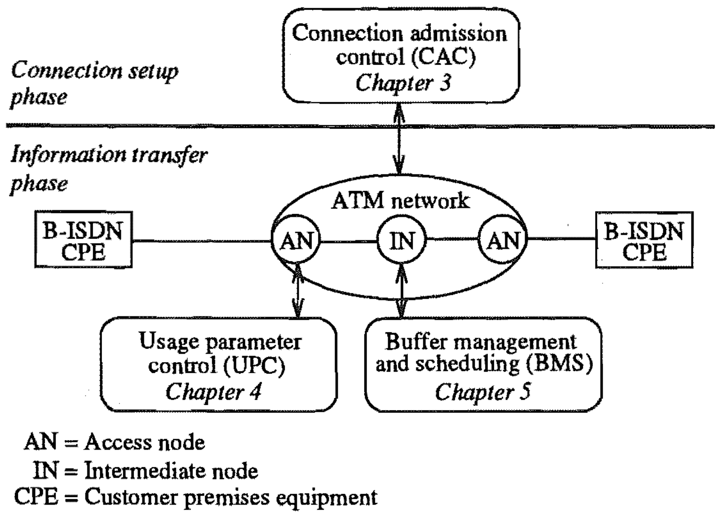


Figure 1.13 CCITT traffic management framework.

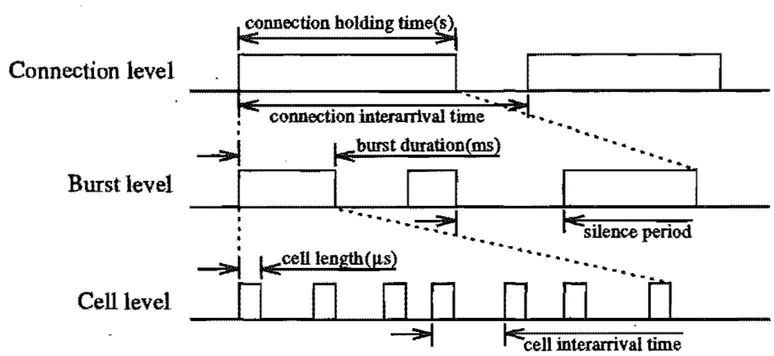


Figure 1.14 Multilevel structure of ATM traffic streams.

(or the peak rate) are required by the network in order to accurately predict its ability to maintain a certain performance level before granting or rejecting a request for a new connection. The required set of parameters is referred to as *traffic descriptors*. The traffic descriptor requirements should be [Roberts and Virtamo, 1991]

- unambiguous, i.e. it should define the traffic characteristics;
- understandable to the user, who must declare their values and capable of interpreting consequential network reaction, such as connection charging;
- sufficient to determine whether or not a connection can be admitted by the network without provoking congestion;
- controllable/enforceable by the network to ensure that a user who declared small values for gaining admittance does not then offer heavier traffic, possibly provoking congestion.

The peak bit rate, average bit rate and burstiness are the most commonly used parameters to meet the above requirements. Among them, burstiness is regarded as one of the most important parameters to describe the behaviour of a specifc traffic source in terms of how densely or sparsely is the stream of arriving cells. It can be defined in various ways and a survey of some of these definitions can be found in [Bae and Suda, 1991]. Among the definitions, the one based on the ratio of peak bit rate to average bit rate is the most often used. However, the definition require additional parameters, since connections with the same peak and mean rates can have very different burst length statistics. This additional parameters should include expected burst length or maximum burst length, since as shown by Gallassi *et al.* [1989] that traffic with long burst length degrades the effectiveness of statistical multiplexing in ATM networks. The peak rate (R_p), mean rate (R_m), and mean burst duration (T) for various traffic sources are listed in Table 1.2 [Sriram and Whitt, 1986; Maglaris *et al.*, 1988; Verbiest and Pinnoo, 1989; Baiocchi *et al.*, 1991].

Traffic source	R_p (Mbps)	R_m (Mbps)	$T(ms)$
ADPCM TASI voice	0.032	0.011	350
Data	10.000	1.000	10
Videophone	10.575	3.900	310
Broadcast video	44.100	16.756	330

Table 1.2 Different traffic descriptors for B-ISDN services.

In addition to the above parameters, the ratio of the peak rate and the link rate is also important. As shown in [Ohnishi *et al.*, 1988; Hughes *et al.*, 1990b; Rasmussen *et al.*, 1991], multiplexing connections with high values of peak to link rate ratio (in the order of magnitude 0.1 or larger) is not efficient, and hence peak bandwidth should be allocated for such connections.

Quality of Services

Each B-ISDN service requires a specific *quality of service (QoS)*, which are commonly defined in terms of end-to-end cell loss rate, end-to-end transfer delay and maximum allowable cell delay variation (CDV) or jitter. CDV measures the alteration in the initial time structure of a cell stream passing through an ATM network. This alteration is due to random delays experienced by the cells in the switching and multiplexing stages. It results in clumping and dispersion effects of cells within the stream [Boyer *et al.*, 1992; Guillemin and Monin, 1992].

The end-to-end cell loss rate requirements specify the maximum admissible cell loss in a connection. Cell loss in ATM networks can be due to uncorrectable bit errors caused by physical layer, forced discarding of cells at the usage parameter controls, buffers overflows due to instantaneous cell traffic overload, and smoothing buffer overflows/underflows at the receiving ends.

The end-to-end cell delay requirements specify the maximum allowable delay and delay jitter experienced by cells in a connection. The delay includes cell assembly time, propagation delays, transmission time, switching and queueing delays in the buffer, and a compensation of cell delay jitter at the receiving ends.

Table 1.3 lists the various QoS bounds used for different mutimedia services [Hehmann, 1990; Ferrari, 1990].

Service type	Acceptable cell loss rate	Maximum delay (s)	Maximum delay jitter (ms)
Voice	$< 10^{-3}$	0.25	10
Standard video quality	10^{-3}	0.25	10
Compressed video	10^{-9}	0.25	1
File transfer	0	1	-
Real-time data	0	0.001-1	-
Image	10^{-9}	1	-

Table 1.3 Different QoS requirements for B-ISDN services.

In addition to cell-level QoS, burst-level QoS has also been suggested for applications with traffic characterised by long bursts, such as large file or image data transfer. The burst-level QoS can be expressed in similar terms as cell-level QoS, namely burst delay and burst loss rate [Roberts, 1991].

1.5.2 Connection Admission Control

The *connection admission control (CAC)* provides the main method of controlling traffic performance in ATM networks. Its objective is to ensure that sufficient network resources are available for each admitted connection. When a request for a new connection is received, the CAC makes the decision either to accept or to deny the connection request.

The simplest CAC approach is the *peak rate allocation* method, which allocates an amount of bandwidth equal to the peak rate of a connection. A new connection is accepted if the sum of the peak rate of all connections sharing the same transmission link is no more than the link's maximum cell rate. Although the approach is simple to implement and to understand, quality of service problems can still arise when connections with different peak rates share the same link [Burgin and Dorman, 1991]. Another obvious drawback of this approach is that it makes poor use of network resources, especially in the presence of bursty traffic as it does not exploit the advantage of statistical multiplexing among the connections. To overcome these drawbacks, other techniques which make better use of the available network resources and provide probabilistic guarantees of service quality have been proposed in recent years. Three such approaches can be identified based on the way a decision is made.

The first approach is called the *virtual trunk* method [Mase and Shoda, 1991; Vakil and Saito, 1991]. It operates in a similar fashion to the bandwidth allocation in synchrononous transfer mode (STM) networks, where a new connection is accepted only if a trunk of the prescribed bandwidth for that class of connection is available.

The second approach is called the *QoS evaluation* method. Given a fixed bandwidth capacity, the method will evaluate the resulting QoS if a connection is accepted. It rejects the new connection if the resulting QoS violates that allowed for a transmission link or a virtual path. The resulting QoS can be based on an average [Dutkiewicz and Anido, 1990; Esaki *et al.*, 1990; Addie and Zukerman, 1993b], an instantaneous [Kamitake and Suda, 1989] or an upper bound cell loss probability [Rasmussen *et al.*, 1991; Saito *et al.*, 1991; Miyao, 1991]. The on-line measurement or calculation of the QoS allows the method to take into account the dynamic and changing nature of the real-time traffic conditions. However, the time constraint requires that the algorithm used be fast though possibly having reduced accuracy.

The third approach is called the *virtual bandwidth* or *equivalent bandwidth* method [Gallassi *et al.*, 1989; Decina and Toniatti, 1990; Jabbari, 1990; Guerin *et al.*, 1991; Monteiro *et al.*, 1991]. This method estimates the effective or the minimum bandwidth required by a connection to support the prescribed QoS levels in a homogeneous traffic environment. It significantly reduces the amount of on-line calculations, which allow more precise calculation mechanisms to be used than with the on-line methods. It has proved to be a popular method and has been widely studied [Vakil and Saito, 1991; Mase and Shoda, 1991]. The algorithms that have been proposed differ in the traffic descriptors used. After determining the equivalent bandwidth, the CAC can make its decision by ensuring that the bandwidth requirement does not exceed the available link bandwidth or some predetermined fraction of available bandwidth [Ilyas and Mouftah, 1990], or available virtual path bandwidth for the class of services [Wang *et al.*, 1990]. In the Chapter 3 of this thesis, the virtual bandwidth method will be adopted for determining the amount of bandwidth required by a connection that requires two different quality of services due to the presence of *pretagged* cells, i.e. cells which are pretagged by users as low priority.

1.5.3 Usage Parameter Control

During information transfer phase, traffic from a connection is monitored by a *usage parameter control (UPC)* or *policing* function at the ingress to the network for compliance to its traffic descriptors or its negotiated traffic parameters. The role of the UPC is primarily to protect the network from incidence of excess traffic due to short term traffic variations, malicious user actions or terminal equipment malfunctions that would overload the network and prevent the network from functioning correctly. It is to some extent similar to the flow control concept, but with sender and receiver working autonomously.

As the UPC is required by each connection and must operate in real time, this means that the mechanism used must be fast, simple, and cost effective to implement in hardware and ideally, it should be transparent for connections that respect the negotiated traffic parameters, and act only on cells violating the parameters. Based on these requirements, there have been numerous algorithms proposed in recent years. Depending on which parameters they police, the policing functions can be classified as peak, average, and distribution based policing [De Prycker, 1991].

Peak policing enforces the source peak rate. It is normally done by checking that the ratio of the maximum number of cells within a time interval to the interval length does not excess the peak rate. Hence its effectiveness is highly dependent on the time interval chosen. Too long an interval can limit the responsiveness of the policing mechanism. A spacer controller [Boyer, 1990; Boyer *et al.*, 1992] is an alternative to the peak policing. The scheme spaces out cells, which arrive too closely, by a time interval equivalent to the inverse of the peak rate.

Average policing for bursty traffic ensures that the ratio between burst and silence period over a time frame or the mean rate is as negotiated. It can only police either the average rate or the maximum burst length but never both [Rathgeb, 1991]. Based on the time span, when the policing action is carried out, average policing can be classified as either implicit time frame or explicit time frame policing [Okada *et al.*, 1991]. With an explicit time frame, the frame is segmented by an explicit frame marking. This class includes (r, T)-smooth stream [Golestani, 1991] and window-based (e.g. jumping window, moving window) schemes [Rathgeb, 1991; Dittman *et al.*, 1991; Hemmer and Huth, 1991]. On the other hand, with implicit time frame, the time frame is derived by the maximum continuous burst length permitted. This class includes leaky bucket scheme and its derivatives.

In addition to policing the first moment of the source bit rate distribution as in peak and average policing, distribution based policing, such as gibarit policing [Joos and Verbiest, 1989; Unteregelsbacher and Mouftah, 1991], uses higher moments of the distribution. However, due to complexity in calculating the higher moments, the implementation of such policing is questionable for meeting high speed and low cost requirements.

Among the policing schemes described above, the general consensus is that the leaky bucket algorithm is the most simple and thus, the most often used mechanism for policing average and peak rate. Although it has been shown to be useful in policing peak rate, the leaky bucket algorithm is less efficient for policing average rate of bursty traffic, such as video and data traffic, since large bucket depth or token rate are required [Rathgeb, 1991]. Therefore our interest is to study the benefits of incorporating marking, buffering and priority control in the leaky bucket algorithm for policing video sources which generate both *untagged* (high priority) and *tagged* (low priority) cells in Chapter 4.

1.5.4 Buffer Management and Scheduling

In ATM networks, most connections will not be carried individually through the network, but will be statistically multiplexed with other connections within the network. The convergence of cells from various connections at the intermediate nodes result in competition for buffer space and link bandwidth, which can lead to cells being discarded freely or delayed excessively, hence degrading the QoS of the connections.

With ATM networks being expected to support a wide range of QoS requirements, one way to provide different QoS is by carefully dimensioning the buffer sizes and link bandwidths to satisfy the most stringent QoS requirements. Such an approach leads to poor utilisation of network resources. A more flexible approach can be achieved by providing some priority mechanisms within the network. Priorities can be of two different types, namely *delay* (or time) priorities and *loss* (or space or semantics) priorities.

Delay priorities [Jaiswal, 1968] provide preferential service to some classes of traffic in order to control their end-to-end delay and delay variation (jitter). It has been widely used in existing buffer management schemes, such as head of line (HOL) priority [Kleinrock, 1976], weighted round-robin scheme [Lazar *et al.*, 1989; Kalmanek and Kanakia, 1990], oldest-customer-first (OCF) and earliest-deadline-first (EDF) disciplines [Chen *et al.*, 1989], longest queue first served (LQFS) [Fan, 1991], rate-controlled static-priority (RCSP) [Zhang and Ferrari, 1993]. Delay priorities can typically be employed only to meet different service requirements among various connections which are not related in time (e.g. synchronised).

On the other hand, loss priorities [Sumita and Ozawa, 1988; Kroner, 1990] provide preferential access to buffer spaces without affecting cell ordering. The priority can be assigned *implicitly* or *explicitly* [Gallassi *et al.*, 1990b]. The implicit policy assigns the priority on a connection basis, hence all cells in a connection will have the same priority level. On the other hand, the explicit policy assigns the priority on a cell basis. This policy allows a source to assign different priority to each cell according to the significance of the information contained in the cell. It also allows a policing scheme to mark cells, which violate the negotiated traffic parameters, as low priority. A large number of buffer management schemes have been proposed recently to implement both the implicit policy, such as separate routing scheme [Kroner, 1990], and the explicit policy, such as partial buffer sharing scheme [Kroner, 1990] and push-out scheme [Hebuterne and Gravey, 1989]. A combination of delay priority and loss priority have also been considered in various publications, including (T_1, T_2) scheme [Sriram, 1990], low delay or low loss (LDOLL) scheme [Awater and Schoute, 1991], push-out with service priority schemes [Mitrou and Pendrakis, 1991; Chen and Akyildiz, 1992; Chen and Mark, 1992; Huang *et al.*, 1993].

Despite its extensive usage in existing buffer management schemes, delay priority has been considered less useful in the ATM context as reduction in queueing delay offered by delay priority only contributes insignificantly to the end-to-end delay, which is dominated by large transmission delays [Eckberg *et al.*, 1989; Rothermel, 1990]. Moreover the choice of small buffers in order to minimise the cell delay variation will also reduce the queueing delay. On the other hand, due to these small buffers, the concern for controlling cell loss becomes more important. Therefore, in the Chapter 5 of this thesis, we shall deal with the usage of loss priority when we investigate the performance of various buffer management schemes in terms of their protection of high priority traffic against the overload from low priority traffic.

1.5.5 Reactive Control

Doshi *et al.* [1990] and Woodruff and Kositpaiboon [1990] suggested the need for incorporating reactive control into an overall control strategy in order to provide robustness and flexibility for ATM networks to accomodate diverse service types. The reactive control can be achieved by using an explicit congestion notification (ECN). The scheme requires a traffic source to throttle back upon receiving the congestion notification from the network.

ECN is commonly suggested for data applications, e.g. LAN interconnection, where congestion events tend to last long enough for ECN to be beneficial [Fowler and Leland, 1991]. One popular example of ECN applications in existing data networks is a solution based on applying the source quench message of the TCP/IP protocol [Stallings, 1991]. In ATM networks, two variety of ECN have been proposed, often referred to as forward ECN and backward ECN. In the forward ECN mechanism a congestion indicator is borne within the header of ATM cells. It is set by a congested network element and send to the destination end, which in turn forwards the indication to its traffic source. The forward ECN was investigated by Makrucki [1991] and found to have significant value in helping end-to-end protocol adapt to the state of the network, even when the propagation delays are long. On the other hand, in the backward ECN mechanism the source is informed about the congestion directly by the congested network element, by means of network-element-originated ATM cells. The backward ECN was investigated by Newman [1993] for applications in ATM LANs.

1.6 The Contributions of this Thesis

The focus of this thesis is on the call control and traffic management aspects of B-ISDNs. Unlike other research approaches which treat the issues separately and study them primarily from the network point of view, in this thesis we consider the issues together, following a novel approach that we refer to as *user-network oriented* approach. This approach opens the possibility for involving user terminals in the control process, in order to make use of the user knowledge about the required services.

In the area of call control, the approach allows user terminals to control call establishment processes. On the other hand, in the area of traffic management, the approach allows users to police their cells properly and to selectively tag as low priority either cells containing less essential information and so deemed expendable or cells protected by end-to-end error recovery schemes. Based on the existence or absence of these pretagged cells within a connection, we differentiate connections as *pure* (without pretagged cells) and *mixed* (with pretagged cells).

The aims of this thesis is to investigate the benefits of such an approach in terms of increased flexibility for establishing various services, minimised bandwidth requirements at the user ends and maximised resource utilisation within the network, yet satisfying the QoS requirements.

The major contributions embodied in this thesis can be summarised as follows.

- Development of a new call model and a new signalling architecture in Chapter 2, which includes an additional party control layer and separate connection control protocols for point-to-point and multipoint connections.

- Two new methods for allocating bandwidth for mixed connections in a homogeneous traffic environment and one method for a heterogeneous traffic environment are proposed in Chapter 3. These methods require a single search for the bandwidth and exploit the statistical multiplexing of untagged and pretagged traffic in the connections.

- Four modified leaky bucket schemes for policing mixed connections are proposed in Chapter 4 along with discrete-time analytical solutions for evaluating their performances.

- Proposal and analysis of a dual queues with limited cyclic service (DQCS) scheme in Chapter 5 for maximising resource utilisation while protecting traffic of pure connections from overload by low priority traffic of mixed connections when the traffic converges at ATM switches.

Other noteworthy contributions made in this thesis include

- Proposal in Chapter 2 of additional capabilities for a multicast switch to allow the switch to support some common conference call functions.

- Performance comparison in Chapter 3 of three recently proposed procedures for matching the characteristics of superposed IPP sources to those of a single MMPP source.

- Proposal in Chapter 4 of two ways for classifying existing leaky bucket schemes, which help in uncovering new schemes for policing mixed connections.

- Proposal in Chapter 5 of a new protection criterion, which allows comparison of the levels of protection of both shared buffer and separate buffer policies.

Chapter 2

USER-NETWORK ORIENTED CALL CONTROL

Since the introduction of data communications, we have seen increasing customer demand for a certain degree of control over their use of public networks. Initially this demand is manifested in the control of information semantics, such as transmitting data over the telephone network using modems. This type of control has gained momentum with the deployment of ISDN and the growing number of multi-service terminals as opposed to single-service ones. Within the terminals, customer control is manifested in the way information from various services being multiplexed into two 64 kbps B-channels provided by ISDN [Nicolaou, 1990; Eigen, 1990]. For example, in the personal multimedia teleconferencing terminal [Nakamura *et al.*, 1990], voice signals are compressed to 32 kbps by an ADPCM[1] codec to share a single 64 kbps channel with graphics signals while video signals use the second 64 kbps channel. Due to the channel rate limitation, video signals have to be compressed and only a small size screen can be transmitted.

In addition to the control of information semantics transported by communication channels, further demand for customer control over multiparty communications, as opposed to two-party communications, has emerged recently in the area of multimedia conferencing. Examples include *computer supported collaborative work (CSCW)*, which aims at assisting group cooperative work through teleconferencing [Bowers and Benford, 1991], and *distributed multimedia information system (DMIS)*, where a call is initiated to obtain database information from several independent sources [Little and Ghafoor, 1991a]. The customer control of multiparty communications is presently achieved through *private automatic branch exchange (PABX)* arrangements. However, these arrangements are limited by the software built into the PABX and have limited ability to observe and control the states of individual participants.

The software limitation is being resolved with the recent convergence of the computer and telecommunication industries, leading to the increasing use of workstation based multi-service terminals. As computing technology has long been the basis for telephone switching and network management, it is also possible to use the workstations to direct the steps of call processing. This approach is generically known as the *switch-computer applications interface (SCAI)* in contrast to the *intelligent network (IN)*, which is network-oriented approach. MERMAID[2] [Maeno *et al.*, 1991] and MIAS[3] [Coolegem *et al.*, 1991] are two examples of workstation based conferencing systems operating in the present network environment. In recent times, the popularity of this approach has been boosted by the improvement in workstation processor at a rate between 1.5

[1]ADPCM stands for adaptive differential pulse code modulation.

[2]MERMAID: Multimedia Environment for Remote Multiple Attendee Interactive Decision-making.

[3]MIAS: Multipoint Interactive Audio-visual System.

and 2 times per year [Hennessy and Jouppi, 1991] along with the advent of ATM networks.

In this chapter, we propose a customer based call control structure, which aims to make use of the flexibility and the multicasting capability of ATM networks. The call control supports generic multiparty multimedia services, such as conference services, by using either point-to-point or multipoint connections. The approach taken for supporting multipoint connections within the network is discussed in Section 2.1. It is followed by a proposal of additional capabilities in a multicast switch for supporting some common conference call functionalities. Section 2.2 describes a call model for generic multiparty multimedia services. Unlike existing call models [Minzer, 1991; Gaddis *et al.*, 1992], which have a unified model for point-to-point and multipoint calls, our model differentiates the call model into two types, namely *basic model* for point-to-point calls and *enhanced model* for multipoint calls. Based on the model, in Section 2.3 we propose a new signalling architecture and describe the functionalities of each layer of its associated call control structure along with the B-ISDN signalling requirements. Section 2.4 provides examples of the control structure usage in establishing the most complex calls, such as video-conference calls. The impact of the call control structure on simple call establishment is discussed in Section 2.5. Analytical results are provided to illustrate the advantages of the call control structure in reducing the call establishment delay for this type of calls. Section 2.6 concludes this chapter by summarising advantages of the proposed call control scheme as compared with existing ones.

2.1 Multipoint Connections

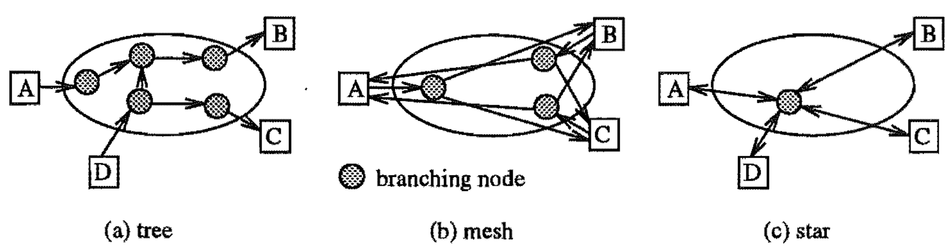


Figure 2.1 Configurations for multipoint connections.

A number of multipoint configurations have been considered for supporting multipoint connections in ATM networks, including tree [Bubenik *et al.*, 1991; Ong and Schwartz, 1991], mesh and star [Ly *et al.*, 1991; Newman, 1992] configurations (see Figure 2.1). The tree configuration allows optimised utilisation of network resources through a number of branching nodes within the network. However, the process of establishing these branching nodes and controlling them can be very complex for calls with a large number of parties. On the other hand, the mesh configuration offers the simplest arrangements of connections by using a point-to-multipoint connection for each information type from each party to all other parties. This approach, however, consumes a large number of VCIs and incurs a great deal of administrative overhead when parties are added to and deleted from the call [Newman, 1992]. Furthermore, it does not represent an efficient use of network resources. A more efficient alternative can be achieved by using a star configuration, while keeping the complexity low. This alternative has been recommended by [CCITT, 1992c].

In star configuration, connections are established from each party to a central node, which redistributes information to all parties (broadcast) or to a subset of parties (multicast) in a call. The connections can be of virtual channels or of virtual paths. Assuming that virtual channels (VCs) are used, a connection between two parties A and B can then be viewed as comprising an input VC, carrying information from party A to the central node, and an output VC, carrying information from the central node to party B. Cells destined to a party from all other parties are multiplexed into a single output VC for each information type. Such multiplexing can reduce the bandwidth requirements through exploiting statistical variations of the traffic sources. However, this does not come without cost, as it raises the possibility of contention among the cells destined for the same output VC. Further requirements arise due to the fact that the destination party may not want information from all other parties at the same time. So, additional capabilities are needed at the central node to meet these requirements by controlling the mapping of information from a single input VC to a number of output VCs and from several input VCs to a single output VC. These issues are discussed in the following sections with the focus on multimedia conference services.

2.1.1 Information Mapping

Information mapping plays a key role in conference services to facilitate two common functionalities, namely

- suppression, which allows temporary suppression of conversation between two parties from a group of parties in the call or temporary separation of subgroups for private chatting.
- selective viewing, which allows a destination party to selectively view or access information from a particular party or a group of parties

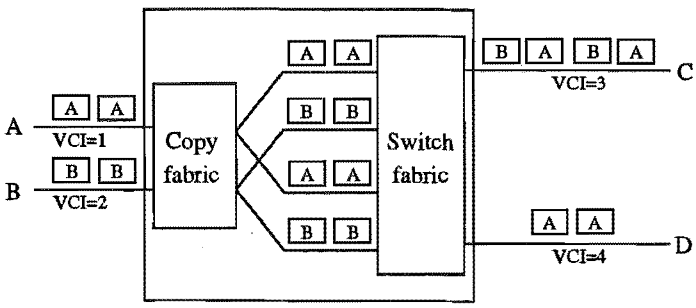
The second functionality is also commonly required by broadcast services, where a number of copies of information, as for example TV programmes, are offered to subscribers by the service providers and the subscribers can view one or more programmes either alternately or simultaneously.

A way to facilitate these functionalities is by modifying the state of the call. For example when a new window is opened to view another party during a conference, users can make a request for additional bearer channel to support the video channel or increase the available bandwidth on existing bearer channel [Jeffrey and Sarma, 1992]. However, such an approach requires a great deal of administration and signalling due to the need for modifying the state of the channel, the number of cell copies generated by the copy fabric and the routing entries of the *cross connect table (CCT)* in a broadcast switch, which implies that the approach is obviously unsuitable if viewing states are allowed to change dynamically.

An alternative, which avoids the need for modifying the state of the call, is by having a *multipoint control unit (MCU)*, such as conference bridges, at the central node. MCUs make use of higher layer functionalities to select, process and distribute multimedia information to all conferencing parties [Minzer, 1991; Ly *et al.*, 1991]. However, the information processing at higher layers implies additional delay which may be detrimental for real-time video services. Furthermore, this approach requires a large initial cost to provide the special resources.

The drawbacks of current approaches motivate us to consider a more efficient approach by supporting the functionalities directly within a multicast switch (refer to Figure 1.8). This can be done by adding additional fields to the switch's CCT in order to control the routing of the cells from the copy fabric to the switch fabric without actually modifying the required number of cell copies or the routing entries of the CCT. The functionalities operate on specific VC only and independently of other VCs in a call.

These additional CCT fields are *permit* and *access* fields to facilitate, respectively, suppression and selective viewing functionalities. An operation of the resulting multicast switch is illustrated in Figures 2.2 and 2.3, which, respectively, show the switch's initial and current conditions.

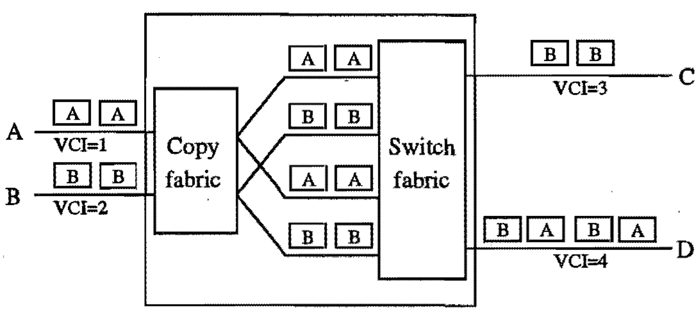


(a)

VPI/VCI in	VPI/VCI out	permit	access
1	3	1	1
1	4	1	1
2	3	1	1
2	4	1	0

(b)

Figure 2.2 Initial multicast switch condition (a) and its CCT (b).



(a)

VPI/VCI in	VPI/VCI out	permit	access
1	3	0	1
1	4	1	1
2	3	1	1
2	4	1	1

(b)

Figure 2.3 Current multicast switch condition (a) and its CCT (b).

The figures show a many-to-many connection between users A, B and users C, D. The connection consists of two input VCs and two output VCs. The initial CCT condition (Figure 2.2(b)) is set during connection establishment. Initially permit fields for all output VCs connected to an input VC are set to 1, which indicates that resulting copies of ATM cells from the input VC are to be passed to all output VCs. The field values can be changed dynamically during the existence of the input VC. Prior to sending sensitive information, users transmitting information via the input VC can reset the permit fields for specific output VCs to 0 by including a control cell at the beginning of the cell sequence, which are to be suppressed. A field value of 0 causes the resulting copies of ATM cells, which are destined to the output VCs, to be discarded. In other words, the users who are at the receiving ends of the output VCs will not be able to gain access to the information. The permit field of the output VCs will remain 0 until another control cell is sent by the transmitting users to set the value to 1. For example, in the current switch condition (Figure 2.3(a)), the information from user A has been temporarily suppressed from user C. Thus, the permit field from VCI=1 to VCI=3 is 0 (see Figure 2.3(b)).

Unlike the permit field, the initial access field values of output VCs are not always 1. Their values are set by the parties who establish the output VCs. The CCT for the initial switch condition (Figure 2.2(b)) indicates that initially user C will have access to information from both users A and B whereas user D will be able to access only information from user A. During the call holding time, the access field can be set and reset dynamically by the users at the receiving end of the output VCs. For example, in Figure 2.3, user D has set the access field from input VC with VCI=2, in order to view the information from user B. Due to the bandwidth limitation of the output VC, the resulting QoS may degrade with additional information from user B. Therefore, it is entirely up to the discretion of user D to control the information presentation. If the degradation is non-acceptable, then user D can choose either to reset the access field or to request for additional bandwidth for the output VC. This last resort of modifying the channel bandwidth can be avoided, for example, by means of layered coding of voice and video signals.

With the layered coding approach, a source can differentiate coded information as essential and non-essential, and transmit them in high priority and low priority cells, respectively. For a video service, the essential information can include low resolution video components, while the non-essential information can include (stereo) sound, subtitling and teletext [Verbiest *et al.*, 1988]. Returning to the switch example of Figures 2.2 and 2.3, if both users A and B use layered coding, then initially user D needs only request enough bandwidth to accomodate all high and low priority cells from user A. If user D tries to access additional information from user B without increasing the channel bandwidth, then user D may only receive all high priority cells and some low priority cells from both users. This depends on the composition of high and low priority cells and on the buffer management schemes employed at the switches. Further details of the bandwidth allocation for connections carrying both high and low priority cells and the buffer management schemes can be found in Chapters 3 and 5, respectively.

2.1.2 Contention Resolution within Multicast Switch

Protocol Description

Apart from facilitating viewing selection by a receiving user, the access field also has the potential for reducing the contention within the switch, which occurs when cells from several input ports try to access the same output port. In other words two or more users try to talk to a single user at the same time.

Currently a number of contention resolution based on input queueing has been proposed [Hui and Renner, 1990; Chen and Hayes, 1992]. Among them, *one-shot scheduling* is of particular interest for keeping synchronisation among the copies of cells. The discipline requires all copies of the same cell to be transmitted in the same time slot. If contention occurs in one of the output port, then all cells will be blocked. A modification of this scheduling, called *revision scheduling*, has been proposed by Chen and Hayes [1992] in order to improve delay-throughput performance of the discipline. The revision scheduling first selects cells according to one-shot discipline until all remaining cells interfere with the selected cells. It then relaxes the restriction of the one-shot rule to allow individual copies of the remaining cells to content for the remaining free output ports. The scheduling was shown to be the most efficient scheme for input queueing.

Both one-shot and revision scheduling assume that all cells are delivered to the output ports. However, with the introduction of the access field, this assumption is relaxed as we allow cells from an input port to be dropped within the switch if the access field of the output port is reset. Such viewing selection can be incorporated into present scheduling techniques, where it can be carried out prior to scheduling. To see the effects of incorporating viewing selection in revision scheduling, a simulation model of the multicast switch is developed by using DESC++ simulation package [Hartanto, 1993; Hartanto *et al.*, 1994a].

In the simulation model, we consider a switch of size $N_{in} \times N_{out}$ ($N_{in} \leq N_{out}$) with a constant copy number of N_{out}, which represents an extreme load condition as compared to the case with random copy number. Such switch condition is common for broadcast services, where N_{in} programmes are offered to N_{out} subscribers. The service providers have to provide all programmes to users who have subscribed to them, even though the subscribers may only view one or more of the programmes at one time.

One simplest way to provide all the programmes to the users is either by using separate channels or by multiplexing them into a single channel and then letting the subscribers choose the programmes at their premises. Under a multiplexing approach, revision scheduling, which is equivalent to one-shot scheduling for a constant number of copies being equal to the number of output ports, can be used to resolve the contention for the output ports. However, this approach will result in a large bandwidth requirements for each subscriber. An alternative which uses viewing selection allows a programme selection to be done through mapping at the multicast switch, where all the unwanted information/cells will be discarded instead of being mapped to the output ports.

Simulation Results

Simulations are carried out by assuming a Bernoulli cell arrival process at the input ports. The length of each simulation run is controlled by using the *Spectral Analysis* method [Heidelberger and Welch, 1981; Pawlikowski, 1990], which stops the simulation automatically when the required precision is reached. We required the results to have at least 0.05 precision[4] at 95% confidence levels. The results, which are the mean cell delay against throughput performance measures, are shown in Figure 2.4 for a broadcast switch with $N_{in} = 4, 8, 16$ and $N_{out} = 32$. The figure shows the results for the revision scheduling with viewing selection (RS-VS) and without viewing selection (RV-NVS). For the RS-VS, the programmes to be viewed simultaneously by a user has been assumed to be randomly chosen with the maximum viewing selection (N_v) being limited to four.

As shown in the figure, the use of viewing selection reduces the mean cell delay at the input queue and increases the throughput before saturation by shedding the cells from the programme which are not to be viewed by users. This reduction in delay is obviously desirable for real-time

[4]Precision of the final estimates is defined as the relative width of final confidence intervals, i.e. the ratio of the halfwidth of confidence interval and the current value of the point estimate.

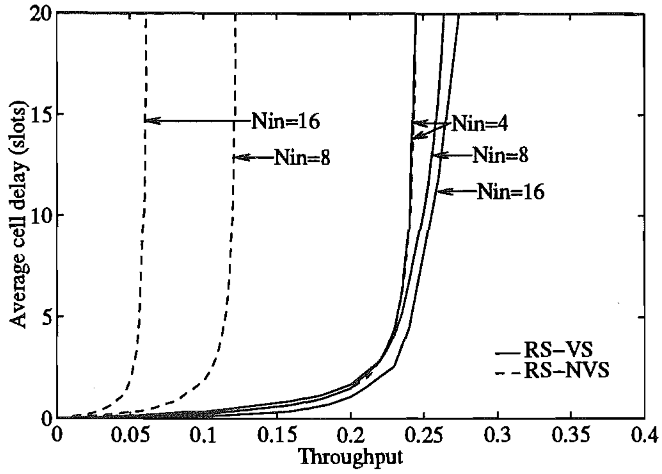


Figure 2.4 Comparison of the delay-throughput performance ($N_{out} = 32$).

video services. One peculiar observation from the graph is that unlike the performance for the RS-NVS, which degrades as the number of programmes or input ports N_{in} increases, the performance of the RS-VS actually improves as N_{in} increases. This is due to the limitation of the number of programmes that may be viewed simultaneously and the random viewing selection by users, which allows the load to spread wider as the number of input ports increases, hence reducing the number of users choosing the same programmes.

The variation of the number of output ports has been found to have no effects on both schemes. This is due to the assumption that the output ports are uniformly selected. On the other hand, the variation of the maximum number of programmes (N_v) that can be viewed simultaneously by a user has a profound effect on the delay-throughput performance as shown in Figure 2.5, while the performance improves as N_v decreases. When $N_v = 1$, we expect that there will be no contention at all. The performance improves further as the number of input ports increases from 4 to 16.

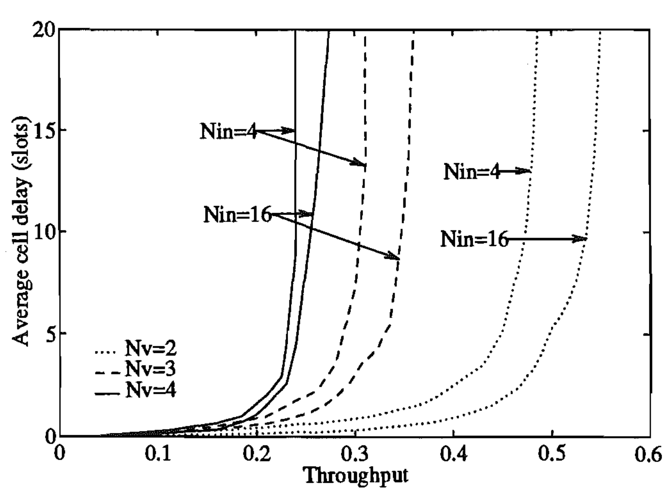


Figure 2.5 The effect of varying the maximum number of programmes to be viewed ($N_{out} = 32$).

2.2 Object-Oriented Model of a Call

An object-oriented model of a call is useful for designing a software implementation of a call control protocol. In this model, a call is viewed as a collection of two or more communicating parties with one or more logical connections associated with each pair of parties. The model is dynamic in the sense that the number of parties, connections and characteristics of the connections are allowed to change dynamically during the call holding time. It also supports the relationship (e.g. synchronisation) among the connections.

The model, proposed in [Hartanto *et al.*, 1992; Hartanto and Sirisena, 1993b], describes a call by using three basic components or objects, namely *Call*, *Party* and *Connection*. The objects are related in a hierarchical way with the *Call* object at the top of the hierarchy. Each object has an *identifier (ID)* or a *pointer (PT)* associated with it, which is provided by the network or by the customer call control, respectively, to allow the object to be manipulated independently.

In defining the objects, we differentiate between *basic* and *enhanced* objects for modelling point-to-point calls and multipoint calls, respectively. The enhanced objects can be derived from the basic objects with some additional attributes.

2.2.1 Call Object

A *Call* object, identified by a Call_PT, represents a B-ISDN call. The attributes of a basic *Call* object includes

- Call_ID, which is provided by the network. The purpose of the call reference/identifier is to identify the call at the local exchange.

- Call type (standard or customised, point-to-point or multipoint). Customised calls are processed and managed by customer call control as opposed to standard calls, which are managed by the network.

- A *PartyList* object, which is a concatenation of Party_PT of all participants or *Party* object in the call.

In addition to the basic attributes, an enhanced *Call* object should include the following attribute

- Call accessibility (open, closed or restricted). An *open/closed* call indicates the permission/restriction for participants in the call, apart from the calling party, to invite other parties to join the call or to add connections to other participants in the call. On the other hand, a *restricted* call means that only a subset of all participants designated by the call owner is allowed to do so. All invitations for new parties or additions of new connections must be verified by the calling party, who may reject the inclusion of the new parties or the new connections. Open and restricted accessibilities are particularly useful in setting up interactive services, whereas close accessibility is useful in setting up broadcast services.

2.2.2 Party Object

A *Party* object, identified by a Party_PT, represents an end-to-end association between two parties. These parties may include user terminals, service providers (e.g. CATV) or network entities (e.g. multicast switches, communication bridge or gateways to other networks). This object allows a simplified relationship between a call and its connections. A *Party* object should include the following attributes.

- Party_ID, which is assigned by the local exchange of the originating party, who establishes a *signalling virtual channel (SVC).*
- Source and destination party addresses.
- Essential attribute, which indicates the significance of a party in a multiparty call. Any failure in inviting an essential party will result in termination of the call.
- A *ConnectionList* object, which lists all connections belonging to a party, plays an important role in controlling multi-connection calls where connections can be added or removed from a call dynamically. It also provides the relationship among the connections, where sublists can be created for connections which require synchronisation.

2.2.3 Connection Object

A *Connection* object, identified by a Conn_PT, represents an end-to-end connection between two parties for carrying information. A basic *Connection* object should include

- Conn_ID, which is provided by the network to identify the user transport channel. It can simply be VPI/VCI value.
- Connection type (VC or VP, uni-directional or bi-directional, point-to-point or multipoint) and its identification (VCI or VPI). The assignment of VCI or VPI in bi-directional connections can be the same or different.
- Connection characteristics in terms of service class, bandwidth requirements (peak rate, average rate and burstiness) and QoS requirements.
- Type of information coding scheme used.
- Essential attribute, indicates the vitality of a connection to a party. If an essential connection cannot be completed, then the party will be excluded from the call. In turn, if the party is essential to the communications, then the call will be terminated and its corresponding *Call* object deleted. Due to the hierarchical structure, this also means the deletion of all *Party* objects in the *PartyList* object associated with the *Call* object and, in turn, the deletion of all *Connection* objects associated with the *Party* objects. The advantages of the hierarchical relationship of call components provided by the model are clear.

In addition to the above attributes, an enhanced *Connection* object should include

- Connection accessibility (open, closed or informed). An *open* connection permits other participants in the call to obtain information from the connection; A *closed* connection implies that only participants authorised by the transmitting user are allowed to obtain information from the connection and an *informed* connection indicates to the network to inform the originating user in regard to the changing of viewing or mapping status to the connection. This last accessibility is useful in broadcast services, e.g. for broadcasters to collect customer rating of the offered programmes.

- *Connection Permit List* contains the list of connections which are allowed access to the information in the connection. This list basically represents a multipoint configuration of the connection. If the list is empty, i.e. no user is allowed access to the information, then the connection will be torn down.

- *Connection Access List* contains a list of connections from which the user can access information from. In the case of unidirectional connection, this list will be empty.

2.2.4 Graphical Call Representations

As mentioned previously, the call model provides a hierarchical relationship among various components of a call. This relationship can be illustrated graphically by using either *block inclusion* or *hierarchical graph* representation. As an example, we consider a three-way video conference call (Figure 2.6) among users A, B and C with all three users being in the voice connection, but only users A and B in the video connection, and users A and C in the data connection. All connections are assumed to be bidirectional.

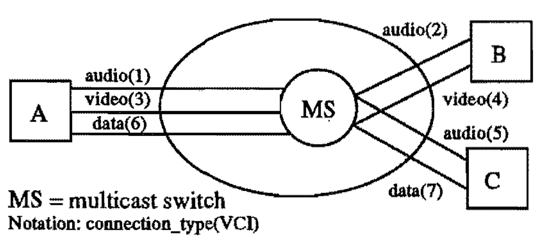


Figure 2.6 Connections in a three-way conference call.

Figures 2.7 and 2.8 show the block inclusion and the hierarchical graph representations for the call, respectively. The arrows in the block inclusion representation indicate the links provided by the *PartyList* and the *ConnectionList* objects. The *S* in the hierarchical graph representation indicates the synchronisation between audio and video connections. Both representations have their own merits. The block inclusion representation is good for representing the logical relationship among various components in a call. On the other hand, the hierarchical graph representation provides a good representation of the physical connections and their accessibilities to various users as listed in the *Connection Permit List* and *Connection Access List* of a *Connection* object.

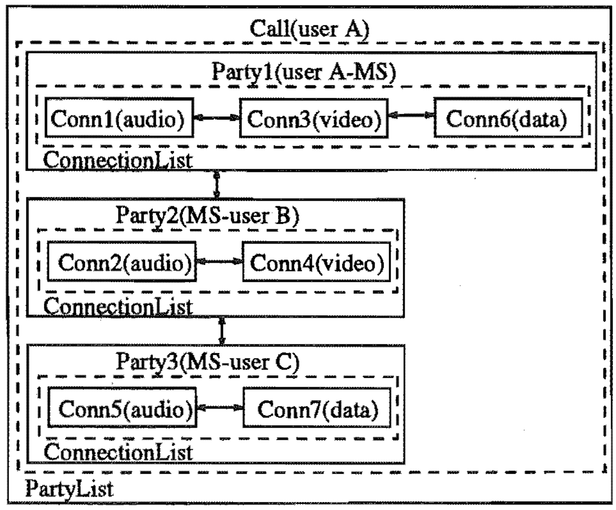


Figure 2.7 Block inclusion representation of a conference call.

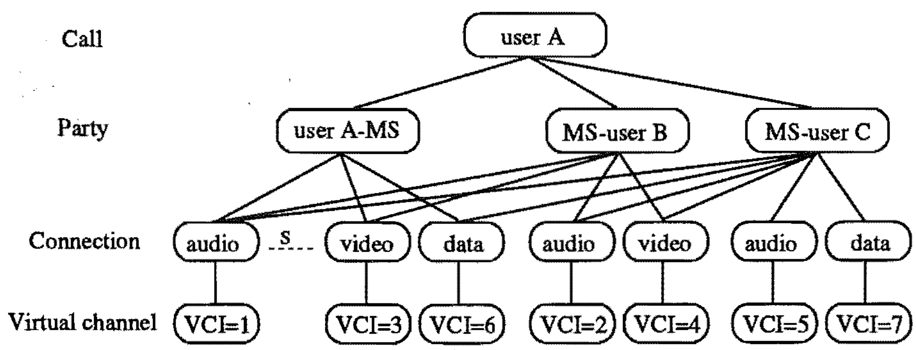


Figure 2.8 Hierarchical graph representation of a conference call.

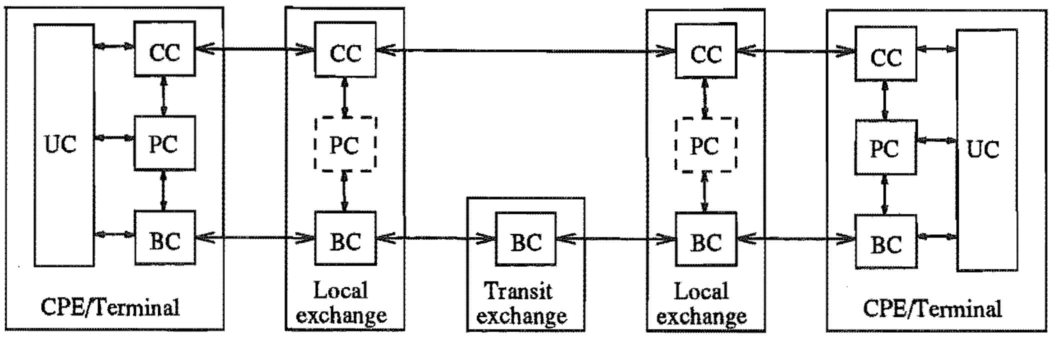
2.3 Signalling Architecture

Based on the call model described in the previous section, we propose a signalling architecture shown in Figure 2.9. Comparing this architecture with the existing architecture of Figure 1.12, we can notice the additional *party control layer*. This layer plays an important part to allow a simpler relationship between call and connection layer and to form a hierarchical structure.

In the signalling architecture, similar layering structure is also proposed for the control at the local exchange in which case the PC layer is optional. The full three layers exist only at the local exchange. However, the two upper layers may also be required in the transit exchange for complex calls, such as conference calls.

2.3.1 CPE Call Control Structure

Each layer in the control structure has its own set of protocol commands, which are used exclusively within the user terminals and can only be called by a higher control layer. The control layer also



UC = User control; CC = Call control; PC = Party control; BC =Bearer/connection control

Figure 2.9 Proposed B-ISDN signalling architecture.

maintains its own database about the call components. For example, the *connection or bearer control (BC)* layer is expected to provide protocol commands for the *party control (PC)* layer and the *user control (UC)* layer for opening a connection. Upon receiving the command, the BC layer will create a *Connection* object for storing all information about the connection and return a Conn_PT to either the PC or the UC layer. The Conn_PT can simply be a pointer to the location of the *Connection* object.

In discussing the protocol commands, the following notation will be used.

- command_name(parameters) —> return_values.

The *command_name* indicates the protocol command provided by a control layer which can be accessed by the control at higher layers and the *return_values* signify a list of identifiers to be returned by the control layer to the requesting layers. The *command_name* without *return_values* means that no identifier is returned to the requesting layers.

Connection Control

The *bearer control (BC)* layer is the most basic layer. Its primary function is to carry out signalling to the network in order to establish a bearer channel or an SVC requested by higher layers. Therefore this layer should provide protocol commands for the upper layers to pass the request, including

- associate(mode, Call_ID, source_party, dest_party, conn_description) —> (Party_ID, Conn_PT) : to request for an SVC and to simultaneously establish a connection between the parties. The mode parameter indicates a point-to-point or a multipoint call setup; the Call_ID parameter is zero for point-to-point call setup and non-zero for multipoint call setup; the conn_description parameter carries the specification of a connection. The parameter may be unspecified, in which case the BC layer will only establish an SVC to the called party and hence the Conn_PT will be zero. Such pre-establishment of an SVC is discussed in Section 2.3.2.

- disassociate(Party_ID) : to request for tearing down the SVC identified by Party_ID.

- open_conn(mode, Call_ID, Party_ID, conn_description) —> Conn_PT : to establish a connection. This command involves creating a *Connection* object for maintaining the database for the connection and returning a pointer to the object (Conn_PT).
- close_conn(Conn_PT) : to close a connection.
- map_conn(Conn_PT1, Conn_PT2) : to request mapping of information from Conn_PT1 to Conn_PT2.

In processing a connection request, the BC layer differentiates two procedures for setting up the connection, depending on the requested mode. For point-to-point connection mode the BC layer simply signals the network to setup a bearer channel. On the other hand, for multipoint connection mode the BC layer also involves in mapping information from one bearer channel to others listed in the *Connection Permit List* and *Connection Access List* of the *Connection* object. In addition, during the connection holding time, the BC layer is also responsible for updating the lists when it receives indication from the network about the establishment of new bearer channels or the termination of any bearer channels in the lists.

Party Control

As can be seen in Figure 2.9, the *party control (PC)* layer does not involve in any signalling to the network. This layer is added to decouple the call control from the connection control. It is responsible for managing the entire communications between two parties from inviting the destination party to controlling the request for connection setup and coordinating the connections that have been established. Its protocol commands include

- open_party(mode, Call_ID, party_description) —> Party_PT : to invite a destination party to join a call. The party_description include source and destination addresses, and a list of all required connections. The party_description may include Conn_PT of a *Connection* object from a two-party single medium call which has been set up without going through the CC layer.
- close_party(Party_PT) : to delete a party from a call.

The PC layer separates all required connections into two groups, *essential* and *non essential* and chooses the most essential connection, such as an audio connection to be established simultaneously with the establishment of an SVC to the destination party by calling *associate* command of the BC layer. If the process of establishing the SVC fails, then the PC layer will fail the whole invitation process and return a *zero* or a *null* Party_PT to the requesting layer. On the other hand, if the process is successful, then the PC layer will try to add other connections to the communications by calling the BC layer's *open_conn* command.

A simultaneous SVC establishment along with the most essential connection is suggested in order to allow the parties to communicate as soon as the call is answered, yet at the same time to prevent wasting network resources by individual connections, which are established before the called parties accepting the call or before the essential connections being able to be set up. As an alternative the PC layer can also set up an SVC without establishing a connection.

Since communications between two parties can involve multiple connections, the PC layer is also required to perform synchronisation between the connections. The synchronisation performed by the PC layer should include intramedia and intermedia synchronisation, i.e. the coordination of the temporal relations within the connection and among the connections, respectively (see Section 6.2.1). In performing synchronisation, the PC layer can form a subgroup of all connections that it has. This opens the possibility for a connection to belong to more than one subgroups. For example, in the communications involving voice, data, and video connections, the PC layer may form a subgroup consisting of voice and video connections and synchronise the connections.

Call Control

The *call control (CC)* layer acts as a call manager for processing a call and for coordinating the activities of various parties in the call. Its protocol commands should include

- open_call(mode, call_description) —> Call_PT : to establish a call with specific establishment mode, viz. standard or customised (point-to-point or multipoint). The call_description may include Party_PT of a *Party* object from a two-party multimedia call which has been set up without going through the CC layer.
- close_call(Call_PT) : to terminate a calll.

The CC layer differentiates procedures for handling standard and customised (point-to-point and multipoint) call. The control will simply pass the call description for a standard call to the network and request the network to set up the call. On the other hand, the CC layer will analyse the call description for a customised call, form party descriptions and request the PC layer to invite the parties. For a customised call in multipoint mode, the CC layer will request for a multicast switch address from the network prior to forming the party descriptions if the address is not known in advance. If the request is rejected, then the CC layer can opt for point-to-point mode or terminate the call.

When inviting the parties, the CC layer differentiates all parties into essential and non-essential group and invites parties from the essential group first, by passing the party descriptions of this group to the PC layer by using the *open_party* command. If the process of establishing communications to these essential parties fails, then the CC layer will initiate the procedure for call termination. On the other hand, if the process is successful, then the CC layer will invite parties from non-essential group. It may retry the process of inviting non-essential party if the first attempt fails. The separation in party invitation reduces the chances of inviting non-essential parties and ends up releasing them due to the failure in inviting essential parties.

User Control

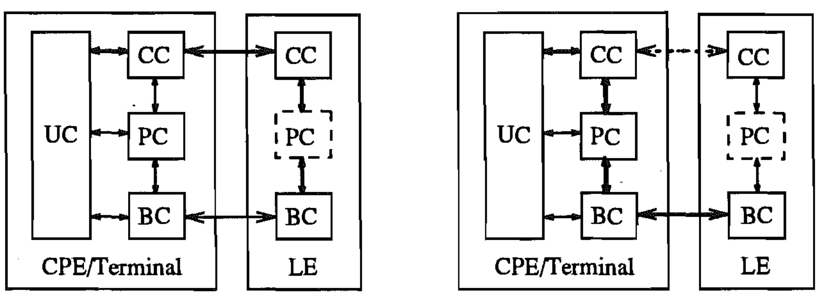
The *user control (UC)* layer forms the human-machine interface between the signalling equipment at CPE and users. The interface can be in the form of a telephone handset, a multi-service terminal, or a workstation. For a workstation-based terminal Screen-Based Technology (SBT) [McNinch, 1990], where windowing software are used to allow features being selected and functions being activated via a mouse, touch-screen or keyboard, can be used to provide simpler and user-friendly terminal.

Using this interface, a call can be established by calling the CC layer's command *open_call*. However, for simple calls users have an alternative to bypass the CC layer. For example, users can set up two-party multimedia calls by accessing the PC layer's command *open_party* and set up two-party single medium calls by accessing the BC layer's command *associate*. This alternative leaves the CC layer to be accessed when setting up multiparty multimedia calls and standard calls only.

When establishing a two-party single medium call by calling the *associate* command, the BC layer will return Conn_PT indicating the connection established. Users can add additional connections to the call by including the Conn_PT along with the descriptions for other connections within the party_description parameter of the *open_party* command. As a result, users will obtain Party_PT from the PC layer. Furthermore, users can add additional parties to the call by including Party_PT within the call_description in the *open_call* command. This alternative illustrates a possible progressive call establishment by users.

2.3.2 B-ISDN Signalling Requirements

The signalling paths in establishing standard calls and customised calls are illustrated by the thick lines in Figures 2.10(a) and (b), respectively. The dotted line in Figure 2.10(b) indicates additional signalling requirements in establishing customised calls using multipoint mode.



(a) signalling for standard calls (b) signalling for customised calls

Figure 2.10 Typical signalling flows for call establishments.

In the following sections we will only discuss some essential protocol commands which should be provided by the call control and connection control layers in the local exchange. These protocol commands will be called by the peer call control and connection control layers at user terminals.

Signalling for Standard Calls

The signalling for standard calls is carried out by the call control layer at the user terminals, where a signal consisting the whole call description is passed to the network. The call control at the local exchange will then process the call according to the steps described in Section 1.4.1 and manages the connections during the call holding time.

This signalling is required to allow users to make use of network-oriented call processing. It is beneficial for some conventional terminals, which do not have the intelligent required to process a call. The protocol commands for this signalling should be provided by the CC layer of the local exchange, which include

- add_call(call_description) —> Call_ID : to request the network to set up a call. The network processes the call and returns a Call_ID to identify the call at the local exchange.
- drop_call(Call_ID) : to terminate the call.

This signalling requires standardisation of a call_description which can include multiple party and multiple connection specifications.

Signalling for Point-to-Point Customised Calls

With this signalling approach, the call control at the user terminals will process the call description, instead of passing it to the network. The signalling is only carried out at the BC layer of the terminal, which implies that the network is not actually aware of the actual composition of the call.

This signalling approach opens the possibility for users to easily reconfigure their calls. Users should always be able to set up calls by using point-to-point connections through having some intelligence in a terminal. The protocol commands for this signalling need only be provided by the BC layer without involvements from the CC layer at the local exchange. Two alternatives for setting up connections at this layer can be identified, i.e. pre-establishment of SVC without setting up any bearer connection and simultaneous establishment of SVC along with an essential bearer connection.

The first alternative is defined in the spirit of CCITT recommendation for call and connection separation. It is beneficial for a connection with large bandwidth requirement. The protocol commands for this alternative should be provided by the BC layer of the local exchange, which include

- request_svc(Call_ID, source_party, dest_party) —> Party_ID : to set up an SVC between the parties without reserving any resources. The Party_ID returned by the network is useful in setting the required connections later on. For multipoint call setup, the network will inform all other parties in regard to the addition of the party if the call accessibility is open.
- release_svc(Party_ID) : to release an SVC from existing call.
- add_conn(Call_ID, Party_ID, conn_description) —> Conn_ID : to add a new connection to a call. If the Call_ID is non-zero, then the network will inform all authorised parties about the existence of the connections.
- drop_conn(Conn_ID) : to remove a connection from a call.
- modify_conn(Conn_ID, conn_parameters) : to modify connection parameters, such as to increase or to reduce the available bandwidth or the QOS of an existing bearer channel.

The second alternative is similar to existing Q.931 ISDN signalling protocol, but with additional capabilities for specifying various connection parameters, such as traffic peak rate and QoS. The advantage of this protocol is that it reduces the delay in starting up the communications between the users. It is suitable for setting up an essential connection with small bandwidth requirements. The protocol commands for this alternative should be provided by the BC layer of the local exchange, which include

- add_party(Call_ID, source_party, dest_party, conn_description) —> (Party_ID, Conn_ID) : to establish an association between the parties, which can be between two end users or between an end-user and a multicast switch, and to set up a connection between the parties simultaneously. This process involves checking the availability of the destination party, performing compatibility check and reserving network resources simultaneously. The process returns a Party_ID and a Conn_ID for referencing the SVC and the connection that has been set up, respectively.
- drop_party(Party_ID) : to delete a party from existing call.

In addition, the protocol commands for this alternative should also include *add_conn*, *drop_conn* and *modify_conn* as defined in the first alternative.

Signalling for Multipoint Customised Calls

For more sophisticated terminals, users can also set up multipoint connections. This requires the network to provide the following protocol commands at the BC layer of the local exchange in addition to the point-to-point signalling protocol.

- map(Conn_ID1, Conn_ID2) : to map information from the bearer channel identified by Conn_ID1 to the one identified by Conn_ID2 and vice versa (bi-directional mapping). In mapping the connections, the network will check the accessibility of the connection to be mapped to if the user requesting the mapping is not the owner of the connection. If it is open, then the mapping will be accepted, and the network will update the copy number of the copy fabric and adding the routing entry between the connections to the CCT within a multicast switch. On the other hand, if the connection is closed, then the mapping will be rejected by the network; If it is informed, then the network will inform the owner of the connection about the mapping, in which case, the owner can decide whether to accept or to reject the mapping.
- suppress(Conn_ID1, Conn_ID2) : to suppress information in Conn_ID1 from being mapped to Conn_ID2 (uni-directional mapping). This command involves only resetting the *permit* field of the CCT without actually modifying the copy number or the routing entry.
- resume(Conn_ID1, Conn_ID2) : to set the *permit* field after the suppression.
- set_access(Conn_ID1, Conn_ID2) : to map information in Conn_ID1 to Conn_ID2 (uni-directional mapping) in order to enable the receiving user to obtain information in Conn_ID1. This involves setting the *access* field of the CCT without actually modifying the copy number or the routing entry.

- reset_access(Conn_ID1, Conn_ID2) : to reset the *access* field of the CCT.

For early implementation of multipoint setup, the address for a multicast switch can be allocated by the network semi-permanently. This overcomes the need for descriptions carrying multiple party addresses. However, the call accessibility can only be closed. In order to offer more flexibility to the users, an additional command for requesting the address of a multicast switch on demand can be included at the CC layer of the local exchange, namely

- request_ms(source_party, authorised_dest_party_list, restricted_dest_party_list) —> (MS_address, Call_ID) : to request the address of a multicast switch. The network will find an optimal location for the multicast switch to all destination parties in order to minimise the distance travelled by the cell copies. If the network can find one, it will return the address of the multicast switch as well as a Call_ID. The network will also inform all authorised destination parties, which are allowed to establish their own connections to the switch, about the switch address and all other parties in the call. The network will not inform the destination parties in the restricted destination party list. In regard to the lists, for calls with open accessibility the restricted_dest_party_list will be empty. On the other hand for calls with closed accessibility, the authorised_dest_party_list will be empty.

- release_ms(Call_ID) : to release the multicast switch and the call. The network will inform all destination parties about the termination of the call.

Evolutionary Steps

In contrast to the signalling for standard calls, where the call_description requires descriptions of multiple parties and multiple connections, we can notice from discussion in the previous sections that most of the protocol commands in signalling for customised calls, except *request_ms*, specify only one connection description as their parameters. The description can be satisfied through a minor extension of the *bearer capability information element* of existing protocol, such as Q.931, to indicate service class, information transfer rate (peak rate in Release 1, average rate and burstiness in Releases 2 and 3) and quality of service requirements. The *Party_ID* and *Conn_ID* parameters can be satisfied by adding new information elements to existing protocol. The need to specify authorised_dest_party_list and restricted_dest_party_list in *request_ms* can also be satisfied quite easily because the lists should only contain the addresses of all destination parties without any other descriptions. The obvious advantage of such evolutionary steps is that it provides backward compatibility.

2.4 Conference Call Establishment and Termination Procedures

2.4.1 Establishment Procedures

In the previous sections, we have identified three signalling alternatives for establishing a call. The choice of particular alternative depends on the level of terminal intelligence and B-ISDN signalling capabilities.

For example, in setting up a conference call, the signalling for standard calls (e.g. three-way calling capability) is a good alternative for terminals with minimal intelligence, while the signalling for customised calls can offer a better alternative for more sophisticated terminals, such as workstation-based terminals. For early implementation of customer call control, multiple point-to-point setup will be ideal, e.g. by using Release 1 signalling capability. If the signalling capabilities for multipoint connection become available, then multipoint setup should be considered in order to reduce the bandwidth requirements by the calls. For demonstrating these two alternatives, we will use the same example of a three-way video conference call as discussed in Section 2.2.4. We assume user A to be the controlling party, which means that user A can affect all resources in the call. The establishment of the call is initiated when user A submits a call description to the CC layer of his CPE by using the CC layer's command *open_call.*

Point-to-Point Setup

With point-to-point setup, the CC layer analyses the call description, creates a *Call* object, and forms a party description for each destination party, resulting in a call model as illustrated by Figure 2.11.

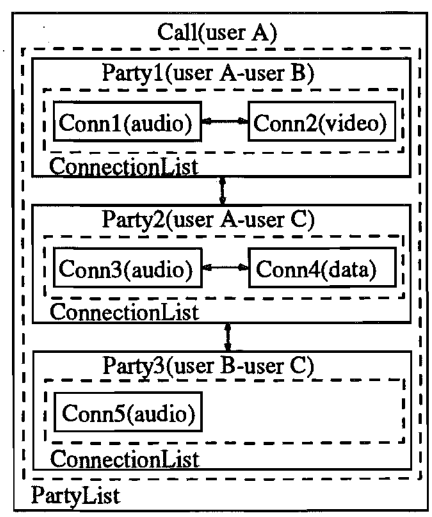


Figure 2.11 Point-to-point conference call model.

The CC layer then separates the parties into essential (e.g. Party1 and Party2) and non-essential groups (e.g. Party3) and sends the descriptions for the essential parties one by one to the PC layer by using the *open_party* command. The PC layer analyses the party description (e.g. Party1), creates a *Party* object, forms descriptions for all connections associated with the party, and groups the connections into essential (e.g. audio) and non-essential (e.g video) types. It then chooses the most essential connection, e.g. audio connection, and requests the BC layer to set up the connection as well as an SVC to user B by using the BC layer's *associate* command.

Upon receiving the request, the BC layer creates a *Connection* object, and sends *add_party* command to the network. The network returns Party_ID1 and Conn_ID1 if it is successful in setting up the connection. The BC layer stores the Conn_ID1 within the *Connection* object and returns both the Party_ID1 and the Conn_PT, which identifies the *Connection* object that it has created, to the PC layer. When the PC layer receives the IDs, it then requests other connections (e.g. video connection) to be established and at the same time sends the Party_PT, which identified the *Party* object that it has created, to the CC layer. When the CC receives IDs for all essential parties, it then tries to invite non-essential parties to the communications.

On the other hand, if the network returns negative response, such as *zero* Party_ID1 or Conn_ID1, due to unavailability of user B or incompatibility in audio coding, or lack of network resources, then the BC layer will delete the *Connection* object and return a *zero* Conn_PT to the PC layer. As the audio connection is essential to Party1, then the PC layer will delete the *Party* object and send a *zero* Party_PT to the CC layer. In turn, as Party1 is essential to the call, the CC layer will terminate the call and return a negative response (a *zero* Call_PT) to the UC layer.

Let us note that, in point-to-point setup, there is no involvement of the CC layer of local exchange in the process, hence there is no call reference (Call_ID) being allocated by the network. In other words, the network actually does not know the whole configuration of the call. Within the network, the call components are simply identified by party references (Party_IDs) and connection references (Conn_IDs). So, the call can be viewed as multiple two-party multimedia calls rather than a single multiparty multimedia call.

Multipoint Setup

With multipoint setup, the CC layer analyses the call description and requests the network to choose a multicast switch connecting to user B and user C by using the signalling command *request_ms*. If the network can provide the switch, it will return the Call_ID1 and the address of the switch to user A. In addition, if the call is either an *open* or a *restricted* one, then the network will also inform all parties (both users B and C), or a subset of authorised parties (e.g. user B only), about all other parties involved in the call. In this example we assume that user B is the authorised party and user C is a restricted party,

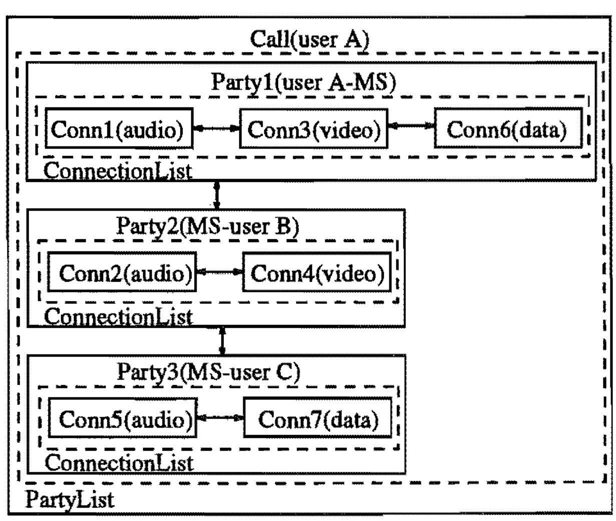


Figure 2.12 Multipoint conference call model.

After obtaining the multicast switch address, the CC will then form the party descriptions, resulting in a call model as shown in Figure 2.12, and pass the most essential party description (e.g. Party1 and Party2) to the PC layer. The PC requests the BC layer to add Party1 with audio connection as the most essential connection.

So far, it is worth noticing the similarities of the establishment procedures with the point-to-point setup. The difference is merely on the destination party address for Party1 and source party address for Party 2 and 3. Further differences occur in the BC layer. When it receives the request from the PC layer, the BC layer will set up a table of physical connection allocations for that call. The table is identified by Call_ID1. The BC layer will then signal the network to establish an SVC and an audio connection between user A and MS. When it receives a positive response from the network, the BC layer will add the Conn_ID from the connection to the table as shown in Figure 2.13 and return Party_ID and Conn_PT to the PC layer.

Call_ID1

	Party1	Party2	Party3
audio	Conn_ID1		
video			
data			

(a) initial condition

Call_ID1

	Party1	Party2	Party3
audio	Conn_ID1	Conn_ID2	Conn_ID5
video	Conn_ID3	Conn_ID4	Conn_ID7
data	Conn_ID6		

(b) final condition

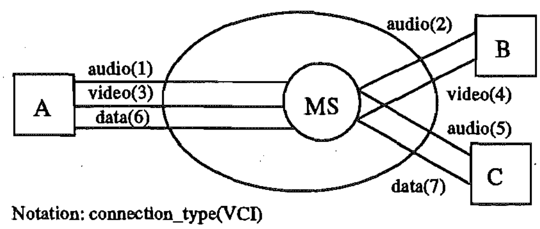
Figure 2.13 Connection allocation list.

Upon receiving positive responses for setting up the SVCs and the audio connections for Party1 and Party2, the PC layer will then use the command *map_conn* to pass the mapping description between the audio connections and at the same time try to add Party 3 as well as video and data connections. When it receives the *map_conn* command, the BC layer will check the connection allocation table. If the connection exists, then it will issue *map* command to the network and update the *Connection Permit List* and *Connection Access List* of the audio connection object Conn1. In processing the mapping request, the network will update the copy number for the input channels at the multicast switch and add a routing entry into the switch 's CCT for mapping the copied cells to the output.

The signalling between the user terminal and the local exchange can be summarised by the following line codes.

request_ms(user_A, {user_B}, {user_C}) —> (MS_address, Call_ID1).
add_party(Call_ID1, user_A, MS, audio_conn) —> (Party_ID1, Conn_ID1).
add_party(Call_ID1, MS, user_B, audio_conn) —> (Party_ID2, Conn_ID2).
map(Conn_ID1, Conn_ID2).
add_conn(Call_ID1, Party_ID1, video_conn) —> Conn_ID3.
add_conn(Call_ID1, Party_ID2, video_conn) —> Conn_ID4.
map(Conn_ID3, Conn_ID4).
add_party(Call_ID1, MS, user_C, audio_conn) —> (Party_ID3, Conn_ID5).
map(Conn_ID1, Conn_ID5).
add_conn(Call_ID1, Party_ID1, data_conn) —> Conn_ID6.
add_conn(Call_ID1, Party_ID3, data_conn) —> Conn_ID7.
map(Conn_ID6, Conn_ID7).

The established connections with the initial CCT mapping are shown in Figure 2.14.



VPI/VCI in	VPI/VCI out	permit	access
1	2	1	1
1	5	1	1
2	1	1	1
2	5	1	0
3	4	1	1
4	3	1	1
5	1	1	1
5	2	1	0
6	7	1	1
7	6	1	1

Figure 2.14 The initial connections and cross connect table of the MS.

As shown in the table, initially there is no communications between users B and C. Since user C is a restricted party, user C will not know the existence of the audio connection to user B. On the other hand, user B is aware of the audio connection and data connection to user C. If at any time user B wants to listen to user C, then she can change the mapping of her audio connection by using *set_access* command. Depending on the accesibility of the connection, if it is *open*, then the network will set the access field for VCI=2 from VCI=5 to 1. Furthermore, if user B wants to set up a data connection to user C, then there are a number of ways this can be achieved, such as user B requests user A to set up the connection through verbal communications or user B sets up a data connection and maps the connection to user C's data connection. With the first alternative, if user A agrees, then he uses *add_conn* command to establish the connection followed by the *map* command. With the second alternative, although user B is not the controlling party, user B can set up the connection because she is an authorised party. In mapping the connection to the data connection of user C, the network will check the accessibility of the connection. If it is *open*, then the network will map the connections. On the other hand, if it is *informed*, then the network will inform user C about the mapping and if the mapping is rejected by user C, then user B is forced to tear down the connection.

2.4.2 Termination Procedures

Call termination in a multiparty environment can occur for different reasons, such as a failure in adding an essential party to a call, a request for call release from an essential party or a controlling party's desire to terminate the call. A request for call release from a non essential party will only affect connections associated with that party.

When terminating a call, the CC layer will request the PC layer to release each *Party* object by using *close_party* command and, in turn, the PC layer will request the BC layer to release the connections associated with the party. Upon receiving the request, the BC layer will send the *drop_conn* command to the network to release the bearer channel associated with the connection.

In releasing the connection, the network will inform the destination party of the connection about the release request and then tear down the bearer connection and add the VCI/VPI of the connection to its spare pool of available numbers, so that the next connection that needs to be setup can use these identifier values. After the release of all connections has been executed, the PC layer will then request the BC layer to drop the party, which is carried out by the BC layer through sending *drop_party* command to the network. At the end of the process, the PC layer will inform the CC layer about the completion of a party release. For point-to-point setup, the call is terminated when all parties have been released. On the other hand, for multipoint setup, the CC layer is requested to send the *release_ms* command to the network after releasing all parties to inform them that the call has been terminated.

2.5 Impact of Customer Control on Call Establishment Delay

In addition to offering more flexibility in establishing a call, as illustrated in the previous section, depending on the processing capability of the user terminals the customer call control also has the potential for reducing the call establishment delay. In order to illustrate this possibility, let us consider a basic call processing model within the local exchange, as shown in Figure 2.15.

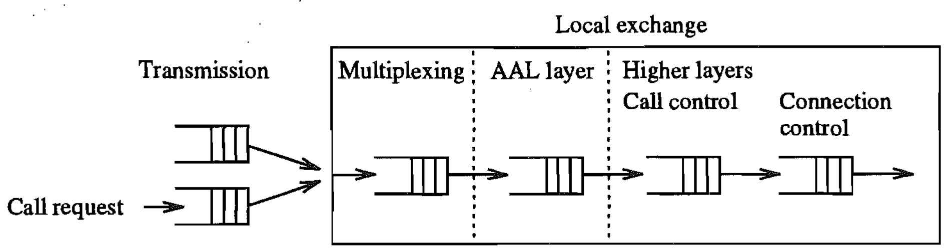


Figure 2.15 Basic call processing model.

From the figure, we can see that the delay in standard call setup includes transfer delay, multiplexing delay, AAL layer processing time and processing time at higher layers. We differentiate the processing time at the higher layers into two components, i.e. call processing time at the call control and connection establishment delay at the connection control. The processing at the call control involves discrimination of various SETUP messages and communications with service control points (SCPs) on how to complete the call setup. On the other hand, processing at the connection control involves routing and resource control in setting up a signalling virtual channel (SVC) and bearer channels.

With the customised call setup, the layered structure of the proposed call control allows simple (two-party single medium or multimedia) calls to be established by calling the PC layer's *open_party* command thereby bypassing the call control at the local exchange and instead directly being processed by the connection control. This implies that any reduction in call establishment delay will be due to avoiding the processing time and delay incurred in queueing for the processor at the call control. Thus, our aim here is to quantify the total processing time (including the queueing delay) at the call control. This can be done by modelling the call control as an M/G/1 queue shown as Call Queue in Figure 2.16.

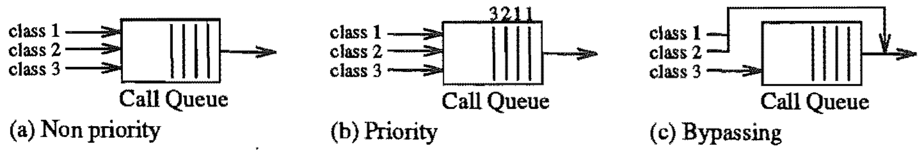


Figure 2.16 Queueing model of a call control.

In analysing the model (see Figure 2.16(a)), we first differentiate three classes of calls: class 1 (two-party single medium), class 2 (two-party multimedia) and class 3 (multiparty multimedia). The arrival of calls to the call control is assumed to be governed by Poisson processes and the calls are processed in first-come-first-served order. The processing time for each class of calls at the call control is assumed to be constant, τ_i (for $i = 1,2,3$) ms. By employing results for the standard M/G/1 queue, we find that the total processing time at the call control (T_i) for class i call is given by the following *Pollaczek-Khinchine mean value formula* [Kleinrock, 1975]

$$T_i = \tau_i + \frac{\lambda\tau^{(2)}}{2(1-\rho)}, \qquad i = 1,2,3 \tag{2.1}$$

where

τ_i = (mean) call processing time of class i at the call control

τ = aggregate call processing time at the call control

$\tau^{(n)}$ = the nth moment of aggregate call processing time distribution

λ_i = mean arrival rate for class i call

λ = aggregate mean call arrival rate

ρ = total traffic intensity at the call control

η_i = probability of a particular call being of class i

and

$$\eta_i = \frac{\lambda_i}{\lambda}, \quad \lambda = \sum_{i=1}^{3}\lambda_i, \quad \tau^{(n)} = \sum_{i=1}^{3}\eta_i\tau_i^n, \quad \rho = \lambda\tau \tag{2.2}$$

As a specific example, we assume that $\tau_1 = 5, \tau_2 = 10, \tau_3 = 20$ and $\rho = 0.8$. Then the total processing time at the call control for various call ratio (η_i) calculated using (2.1) is shown in Table 2.1.

	Non priority			Priority			Bypassing		
(η_1, η_2, η_3)	T_1	T_2	T_3	T_1	T_2	T_3	T_1	T_2	T_3
(1.0, 0.0, 0.0)	15.00	-	-	15.00	-	-	-	-	-
(0.8, 0.2, 0.0)	18.33	23.33	-	12.41	47.04	-	-	-	-
(0.6, 0.2, 0.2)	30.56	35.56	45.56	17.78	37.30	90.99	-	-	21.90

Table 2.1 Total processing time for various call classes.

Comparing the processing time τ_i and the total processing time T_i for class i calls, we notice that the primary contribution to the total processing time is the queueing delay for the call control's processors. This is made worse by the introduction of multiparty multimedia (e.g. conference) calls which require much more intensive processing than simple calls. As the table shows, as the number of such calls increases, the total processing time will also increase. In the non-priority case, we can see that the class 1 calls, requiring 5 ms for processing, may frequently be trapped behind class 3 class requiring much longer processing time, and must therefore wait, on average, nearly twice as long as the queueing delay when no complex calls are present. In order to reduce this queueing delay, one obvious solution would be to provide a higher priority for simple calls enabling them to be processed first (see Figure 2.16(b)). Let us designate the highest priority as 1 and the lowest priority as 3, and assume that within the priority classes calls are served in FIFO order. Then the total processing time at the call control for priority case is given by [Kleinrock, 1976]

$$T_i = \tau_i + \frac{\lambda \tau^{(2)}}{2(1-\sigma_i)(1-\sigma_{i-1})}, \qquad i = 1,2,3 \tag{2.3}$$

where $\sigma_i = \sum_{k=1}^{i} \rho_k$. The total processing times for the priority case are shown in Table 2.1. Of course, it is obvious from the work conservation law (see also [Kleinrock, 1976]) that the reduction in the total processing time for the class 1 calls has come at the expense of the total processing time for calls of class 2 and class 3. However, the doubling of the total processing time for class 3 calls is staggering.

The proposed customer call control provides an alternative which allows calls of class 1 and class 2 to bypass the call control at both the user terminal and the local exchange. This implies that the calls will experience no processing delay at the Call queue (see Figure 2.16(c)), but instead the calls may experience additional delay due to the call processing at the PC and the BC layers of the user terminal. Hence, the net delay reduction that can be achieved through this approach is equal to the total processing time as shown in Table 2.1, plus the processing at the CC layer of the user terminal minus the processing time at the the PC and the BC layers of the user terminal. On the other hand, the delay experienced by the class 3 calls that are still processed by the call control is also reduced, as the aggregate mean call arrival rate will decrease through bypassing. As shown in Table 2.1 the total processing time for the calls is halved.

The given example is an ideal case, where all terminals are assumed to have the processing capacity required to process simple calls. This may not be the case in practice and therefore it is entirely up to the customer to decide whether to process the call at the terminals or to request for call processing by the network. In order to facilitate this decision, future work is required to consider a cost-performance comparison of network-based call processing with per-user call processing.

2.6 Conclusion

In this chapter, we have investigated the possibility of having user terminals control the steps in call establishment processes, with the aim of allowing customers to exploit the nature of their calls, in order to gain flexibility and to reduce the call establishment delay.

To achieve this aim, we introduced a customer call control, which includes an additional party control layer. This additional layer provides a simpler relationship between the call control and connection control layers and, together with these layers, it forms a hierarchical call control structure. The resulting layered architecture allows the flexibility for calls to be set up in various ways, either standard or customised, depending on the call complexities, terminal intelligence and B-ISDN signalling capabilities. The differentiation of the protocols for point-to-point and multipoint communications also allows progressive upgrading of terminal capabilities.

Protocol commands for each layer of the control structure along with the B-ISDN signalling requirements were presented in this chapter. By using the commands we demonstrated some benefits of the proposed call control for establishing both simple and complex calls. For simple (two-party single medium or multimedia) calls, the control structure allows the calls to bypass the call control at the local exchange by accessing the PC layer or the BC layer at the user terminals. Such bypassing has the potential for reducing the call establishment delay for this type of call. On the other hand, in the case of complex (multiparty multimedia) calls, the control structure allows the calls to be supported initially as multiple point-to-point calls. As the signalling capabilities for standard calls or for customised multipoint calls become available, the call control allows the calls to be supported using multipoint connections. Such progressive advancement of terminal capabilities is obviously desirable in order to reduce the initial investment in customer premises equipments.

In order to support some additional functionalities for conference calls using multipoint connections, we also proposed additional *permit* and *access* fields for a multicast switch's CCT to allow dynamic suppression of information by the transmitting party and dynamic change of view by the receiving party without the need to change the state of the call. Protocol commands provided at the BC layer allow users to control the switch functionalities directly. Simulation results showed that the use of the *access* field can also help in reducing the contention within a multicast switch by discarding any unwanted copies of cells, instead of mapping them to the output ports.

In conclusion, the differences between the proposed customer call control and the call control generally described in the literature are summarised in Table 2.2.

	Proposed Call Control	Existing Call Control
Design approach	Evolutionary	Revolutionary
Call model	Hierarchical	Non-hierarchical
Control structure	Three layers: CC, PC and BC layers	Two layers: CC and BC layers
Protocols for point-to-point and multipoint connections	Separate	Unified
Call processing and management	Either by the network or by the users	By the network only

Table 2.2 Major differences between the proposed and existing call controls.

Chapter 3

BANDWIDTH ALLOCATION FOR MIXED CONNECTIONS

In the previous Chapter, we have proposed a customer call control structure, which allows each connection associated with a call to be established separately. In requesting a connection, users are required to present the network with a set of parameters describing the expected traffic characteristics in the connection. The network will then determine the required bandwidth based on these parameters and either grant or reject the request depending on the bandwidth availability.

The issue of determining and satisfying the bandwidth requirements for connections in ATM networks has been widely studied. An equivalent bandwidth concept, as explained in Section 1.6.2, is commonly employed both for homogeneous and heterogeneous traffic conditions. However, most of the methods proposed so far are for connections with single quality of service (QoS) requirement. For example, Dziong and Choquette [1990] estimated the effective bandwidth of a connection using both linear and non-linear approximation; Guerin *et al.* [1991] and Schoute [1988] estimated the equivalent bandwidth of multiplexed traffic by applying fluid flow approximation and Gaussian approximation of superposed teletraffic, respectively; and Gallassi *et al.* [1989] proposed the allocation of bandwidth to each class of connection according to a function of the average and peak rates, and burstiness of the source obtained from a data set pre-computed through simulation. Mase and Shoda [1991] provides a good survey of bandwidth allocation techniques for connections without pretagged traffic proposed before 1991.

With the introduction of cell loss priorities, cells carrying less essential information are allowed to be pretagged already at their sources for transmitting them with low priority. This means that a connection can carry *high priority (untagged)* and *low priority (pretagged)* cells and have two different QoS requirements. As it is beneficial to exploit the special feature of pretagged traffic in order to reduce the total bandwidth requirements, this calls for a bandwidth assignment method that takes into account the presence of pretagged traffic. This issue was addressed by Sriram *et al.* [1991] for layered voice sources, and Pancha and El Zarki [1993] for prioritised MPEG video sources. For more general traffic sources, Saito [1992] proposed a method which allocates peak bandwidth to virtual channels (VCs) containing high priority cells and utilises any bandwidth unused by the high priority VCs for supporting low priority VCs. Unfortunately, the method does not guarantee the QoS for low priority VCs. Gallassi *et al.* [1990b] proposed a method which requires the search for equivalent bandwidths, based on peak rate, average rate and mean burst length, for each QoS requirement separately; the final bandwidth assigned is a linear combination of the equivalent bandwidths. The method is able to satisfy QoS for high and low priority simultaneously, but appears to be time consuming.

Considering the drawbacks in existing methods, in this Chapter we propose analytical methods for determining the bandwidth required by connections carrying pretagged traffic both in homogeneous and heterogeneous conditions which take into account the statistical multiplexing between high and low priority traffic and require only a single search for the required bandwidth.

We begin by classifying connections based on the users' point of view in Section 3.1. Focusing on connections with pretagged traffic or mixed connections, in Section 3.2 we describe the traffic model both for a single connection and for a superposition of connections, which is approximated by an MMPP model. This is followed by a performance comparison of three recently proposed methods for matching the characteristics of the superposed sources to an MMPP model in Section 3.3. In Section 3.4 we analyse the performance of a priority multiplexing system fed by the resulting MMPP model on a discrete-time basis and highlight two important relationships between the loss probabilities of high and low priority traffic. Based on these relationships, in Section 3.5 we propose two new methods for allocating bandwidth to mixed connections under homogeneous and heterogenous traffic conditions, and compare their performance with that of four existing methods. Final conclusions of this chapter are given in Section 3.6.

3.1 Connection Classification

From users' point of view, connections can be differentiated into two classes, namely:

Pure. This class includes all connections from naturally cell loss sensitive services, such as data services, signalling services, CBR voice and video services. All cells in a connection of this class is equally significant and should not be pretagged. Moreover, the traffic of this class of connections should be strictly controlled by the users in order to prevent any cell marking or cell loss within the network.

Mixed. This class includes all connections from naturally cell loss tolerant services, such as voice and video services. These services have the possibility of using hierarchical coding techniques, which code information into layers of different significance and transmit them in cells with high and low loss priorities, e.g. [Sriram *et al.*, 1991] for voice, [Zhang *et al.*, 1991; McLaren, 1992; Pancha and El Zarki, 1993] for video, and [Kim and Mondestino, 1992] for image sources. This results in a traffic stream containing both high and low priority cells. The low priority cells in this class of connections can also be due to the excess cells selectively marked by users (see [Hartanto and Sirisena, 1993a]). These excess cells are considered expendable or else protected by end-to-end cell loss recovery techniques (e.g. forward error correction (FEC) [Ohta and Kitami, 1990]) or cell loss concealment techniques (e.g. pitch replication substitution method [Wasem *et al.*, 1988], simple and motion replenishment method [Jeng and Lee, 1991]).

Further classification of the connections can be based on their traffic descriptors and QoS requirements [Rasmussen and Sorensen, 1989; Awater and Schoute, 1991]. It is preferable to start with as small a number of classes as possible and try to fit new services into existing classes before defining a new one.

Connections of various classes can be multiplexed individually into a transmission link or multiplexed with other connections of the same class into a virtual path. In the latter approach, the boundaries of virtual paths of different classes within a transmission link can be either fixed or movable [Anido, 1989]. A typical routing of connections from two basic classes within an ATM network using the latter approach is shown in Figure 3.1.

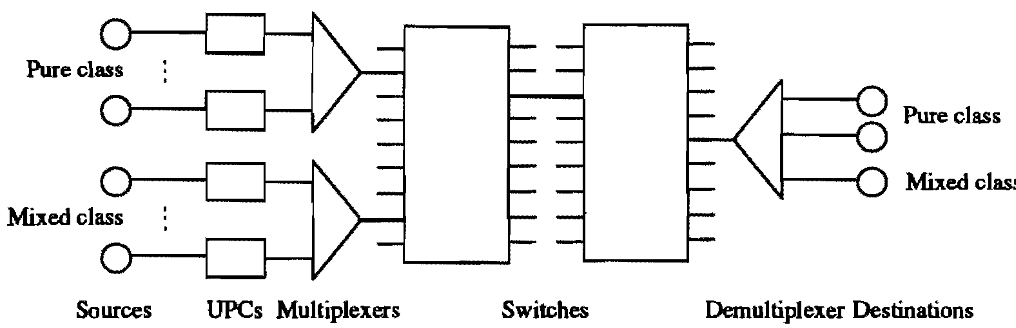


Figure 3.1 Typical connection route within ATM networks.

In the figure, each connection from a user is policed by a usage parameter control (UPC) and the resulting connections of the same class are multiplexed at the access node, either homogeneously or heterogeneously in terms of connection parameters. Further down the network, these multiplexed connections from both classes converge at ATM switches. At these switches, buffer management schemes are used to minimise the interference between cells from both classes of connections.

In the following sections, we will examine in detail the multiplexing of connections at an access node in regard to the bandwidth allocation problem, while leaving the discussion on usage parameter control to Chapter 4 and buffer management within ATM switches to Chapter 5.

3.2 The Traffic Model

In developing a connection admission control, it is necessary to deal with models of teletraffic generated by individual connections as well as by superposition of individual connections, for applying them in the analysis of multiplexers. The following sections discuss these issues.

3.2.1 Single Source

The modelling of an individual source in ATM networks is highly dependent on the type of services being modelled as voice, video, and data sources have different traffic characteristics. The correlated generation of voice cells can be modelled by an interrupted Poisson process (IPP) [Daigle and Langford, 1986; Ide, 1988; Sriram *et al.*, 1991]; the traffic generation from a single data source has been commonly characterised by a Poisson, compound Poisson [Heyman, 1982], train arrival process [Jain and Routhier, 1986] or an IPP [Anick *et al.*, 1982; Rossiter, 1987]; a VBR-coded video source was modelled using linear autoregressive models [Heyman *et al.*, 1992] or using an IPP as a basic element to build the model [Maglaris *et al.*, 1988].

More recent traffic modelling has shown that video and data traffic (e.g. LAN traffic) exhibits a self-similar (or fractal) feature and is better modelled by Pareto distribution [McLaren and Nguyen, 1992; Leland *et al.*, 1993; Garrett and Willinger, 1994]. Although it might have been more desirable to have used this new model for our studies, we have not done so, because the queueing analysis based on this model is not well-established, and hence it would not have been possible to analytically compare the performance of the various control techniques developed here.

We thus adopt an *interrupted Poisson process (IPP)*, shown in Figure 3.2, in our first attempt at analysing the proposed control techniques. This model allows tractable analysis to be carried out when applied to a queueing system. It has been used as a basic element to construct models of various traffic sources [Li, 1991; Kowtha and Vaman, 1992] and is still the most commonly used traffic model [Frost and Melamed, 1994]. This allows us to draw general conclusions about the performance of the newly developed control techniques in relation to other studies of existing techniques without the need for repeating them with a new traffic model.

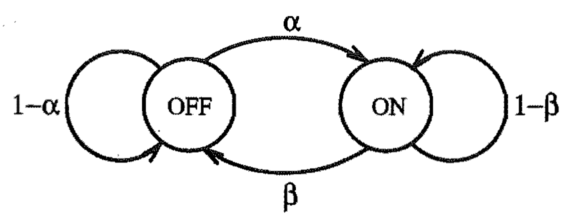


Figure 3.2 Two-state Markov chain models.

In the IPP model, depicted in Figure 3.2, the sojourn times in ON and OFF states are geometrically distributed with averages of β^{-1} and α^{-1}, respectively. When the source is in the ON state, it generates ATM cells at a constant rate λ cells/second. The three parameters α, β and λ, which completely characterise the model, are matched to the statistical characteristics of the cell arrival process for a single source. We assume that each source is characterised by the peak rate (R_p), the mean rate (R_m), both measured in Mbps, and the mean burst duration (T), measured in seconds. The matching of the source characteristics to the IPP parameters is given by

$$\alpha = \frac{1}{T(b-1)}, \quad \beta = \frac{1}{T}, \quad \lambda = \frac{R_p}{l_{ATM}} \tag{3.1}$$

where the burstiness (b) is defined as the ratio of the peak rate to the mean rate, i.e. $b \triangleq R_p/R_m$ and l_{ATM} is the ATM cell payload.

For sources generating both high and low priority cells, the ratio of high priority traffic to the total traffic is denoted by η. In order to simplify the analysis, we assume that the low priority cells arrive independently of high priority cell arrivals. The same assumption has been commonly used in the analysis of priority multiplexing systems [Kroner *et al.*, 1991; Le Boudec, 1991] and it may be justified since we are concerned with the multiplexed sources rather than individual sources.

3.2.2 Superposition of Sources

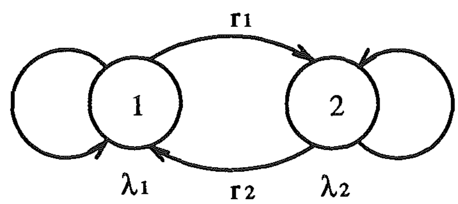


Figure 3.3 Two-state MMPP models.

Several models have been proposed in the literature for representing the superposition of IPP sources. Three main approaches are identified by Baiocchi *et al.* [1991]. Among them, the one that approximates the aggregate arrival process by a suitably chosen simple arrival process is preferred since after matching, the resulting arrival process can be fed to any queue.

In our study, the superposed sources are approximated by a two-state *Markov modulated Poisson process (MMPP)* model as depicted in Figure 3.3. The model is completely characterised by the mean arrival rates, λ_1 and λ_2 ($\lambda_2 > \lambda_1$), of Poisson processes in State 1 and State 2, respectively, and the sojourn time duration at each state, which is geometrically distributed with mean $1/r_1$ and $1/r_2$, respectively. The four parameters of of the equivalent two-state MMPP can be determined by matching certain statistical characteristics (e.g. the mean arrival rate) of the MMPP expressed in terms of λ_1, λ_2, r_1 and r_2 to the corresponding characteristics of the superposed sources.

A wide variety of the statistical characteristics could potentially be matched, therefore a number of methods have been proposed in the literature. Early methods were proposed for matching voice sources by Heffes and Lucantoni [1986] and for matching data sources by Heffes [1980] and Rossiter [1987]. Although these methods perform well in terms of average cell delay, they have been shown to be less accurate for sources with long bursts or high peak rates, both in terms of average cell delay [Liao and Mason, 1989; Lee and Lee, 1992] and cell loss probability [Nagarajan *et al.*, 1991; Baiocchi *et al.*, 1991]. This is primarily due to the the fact that we have overload periods (where the instantaneous cell arrival rate is higher than the output link speed) in the real system, but not necessarily in the approximated system. New matching methods are proposed in [Liao and Mason, 1989; Nagarajan *et al.*, 1991; Baiocchi *et al.*, 1991] to account for this overload state. These methods are based on a division of the arrival rate into an underload period, and an overload period similar to that in [Li, 1988]. On the other hand, Lee and Lee [1992] proposed an improvement to the moment matching method of [Heffes, 1980; Heffes and Lucantoni, 1986] by replacing the matching of the third moment by the decay time constant from the covariance function of cell arrival rate.

In applying the matching procedures for sources generating both untagged and pretagged cells, we first use them to obtain the MMPP parameters for the superposition of sources generating no pretagged cells. We then calculate the percentage of high priority cells η of the resulting MMPP model. For a superposition of N homogeneous sources, η will be the same as η_i ($i = 1, \ldots, N$) of the individual IPP sources. On the other hand, for a superposition of N heterogeneous sources,

η can be calculated as the ratio of total high priority traffic to the overall offered traffic, viz.

$$\eta = \frac{\sum_{i=1}^{N} \rho_i \eta_i}{\sum_{i=1}^{N} \rho_i} \tag{3.2}$$

where ρ_i is the mean offered load of source i, i.e. $\rho_i = (p_i \, \lambda_i / C)$, $p_i = \alpha_i / (\alpha_i + \beta_i)$ and C is the multiplexer link capacity.

3.3 Comparative Study of Three Methods for Parameter Matching

In this study, we choose to investigate the moment matching method proposed by Lee and Lee [1992], the asymptotic matching methods proposed by Baiocchi *et al.* [1991] and [Wang and Silvester, 1993]. We referred to these methods as LL, BBLRW, and WS, respectively. The procedures are the most recently proposed ones, and they offer easy implementations, without the need for inverting Laplace transforms as in [Nagarajan *et al.*, 1991] or for solving simultaneous non-linear equations as in [Heffes and Lucantoni, 1986; Liao and Mason, 1989].

3.3.1 LL Method

This method determines the four parameters of the MMPP source from the mean m, and the variance v of the cell arrival rate, the time constant τ of the covariance function of the cell arrival rate, and the peak-to-mean ratio b of the superposition of the sources. The matching procedure is formulated by Lee and Lee [1992] with the following equations

$$\lambda_1 = m - \sqrt{v/b} \tag{3.3}$$

$$\lambda_2 = m + \sqrt{bv} \tag{3.4}$$

$$r_1 = (1 - \zeta)/\tau \tag{3.5}$$

$$r_2 = \zeta/\tau \tag{3.6}$$

where $\zeta = b/(b+1)$ is the probability of being in State 2.

To obtain the statistics m, v, τ and b of the superposed sources, we note that each individual IPP source behaves as a two-state MMPP source with

$$\lambda_{1i} = 0, \quad \lambda_{2i} = \lambda, \quad r_{1i} = \alpha, \quad r_{2i} = \beta \tag{3.7}$$

Therefore starting from the probability generating function, $g(z,t)$, for the number of arrivals in an interval of length t for MMPP [Heffes and Lucantoni, 1986], we get

$$g(z,t) = \Upsilon \exp\left[\mathbf{R} + (z-1)\Lambda\right] t \, \mathbf{1} \tag{3.8}$$

where

$$\Upsilon = \frac{1}{r_{1i} + r_{2i}} \begin{bmatrix} r_{1i} & r_{2i} \end{bmatrix}, \quad \mathbf{1} = \begin{bmatrix} 1 & 1 \end{bmatrix}^T, \tag{3.9}$$
$$\mathbf{R} = \begin{bmatrix} -r_{1i} & r_{1i} \\ r_{2i} & -r_{2i} \end{bmatrix}, \qquad \Lambda = \begin{bmatrix} \lambda_{1i} & 0 \\ 0 & \lambda_{2i} \end{bmatrix}$$

The mean arrival rate m_i and the covariance function of the arrival rate $\phi_i(t)$ are obtained as

$$m_i = \frac{\lambda_{1i}r_{2i} + \lambda_{2i}r_{1i}}{r_{1i} + r_{2i}} = \frac{\lambda\alpha}{\alpha+\beta} \tag{3.10}$$

$$\begin{aligned} \phi_i(t) &= \Upsilon\Lambda[\exp(\mathbf{R}t - \mathbf{1}\Upsilon)]\,\Lambda\,\mathbf{1} \qquad (3.11)\\ &= \frac{r_{1i}r_{2i}(\lambda_{1i}-\lambda_{2i})^2}{(r_{1i}+r_{2i})^2}\exp(-(r_{1i}+r_{2i})t) \end{aligned}$$

Using the expression for $\phi_i(t)$, we can find v_i and τ_i, [Heffes, 1980], as

$$v_i = \phi_i(0) = \frac{r_{1i}r_{2i}(\lambda_{1i}-\lambda_{2i})^2}{(r_{1i}+r_{2i})^2} = \frac{\alpha\beta\lambda^2}{(\alpha+\beta)^2} \tag{3.12}$$

$$\tau_i = \frac{1}{v_i}\int_0^\infty \phi_i(t)dt = \frac{1}{r_{1i}+r_{2i}} = \frac{1}{\alpha+\beta} \tag{3.13}$$

Finally,

$$b_i = \frac{\alpha+\beta}{\alpha} \tag{3.14}$$

Knowing m_i, v_i, τ_i, and b_i for individual source i, the parameters m, v, τ, and b for the aggregate process of N source can be calculated as

$$m = \sum_{i=1}^{N} m_i, \quad v = \sum_{i=1}^{N} v_i, \quad \tau = \sum_{i=1}^{N}\frac{v_i}{v}\tau_i, \quad b = \sum_{i=1}^{N}\frac{m_i}{m}b_i \tag{3.15}$$

3.3.2 BBLRW Method

Following the definitions in [Baiocchi *et al.*, 1991], let C, measured in cells/seconds, denote the net output capacity from a multiplexer, and $M = \lfloor C/\lambda \rfloor$ indicate the maximum number of sources that can be accomodated in the multiplexer, assuming a peak bandwidth assignment.

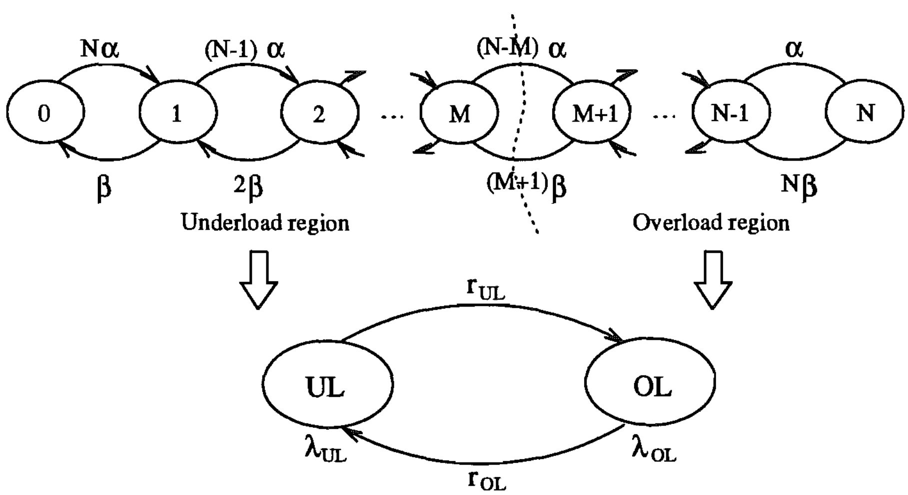


Figure 3.4 Birth-death process and its two-state MMPP approximation.

By assuming that a superposition of N independent IPP sources results in a birth-death process (BDP) with a state transition diagram as shown in Figure 3.4, we can divide the chain of states into two regions, namely the *overload* region, comprising the states $\{M+1,\ldots,N\}$ where the instantaneous sum of source rates exceeds the capacity C, and the *underload* region, comprising the states $\{0,\ldots,M\}$.

Let us indicate the underload and overload regions of the BDP as State UL and State OL, which respectively correspond to State 1 and State 2 of a two-state MMPP model as shown in Figure 3.4. Starting from the fact that the time spent, or the absorption time, in the overload region has a phase-type probability distribution (PH distribution) [Neuts, 1981], Baiocchi *et al.* [1991] approximates the distribution by a negative exponential distribution with mean equal to $1/r_{OL}$. The mean depends on the dominant eigenvalue of the $(N-M)\times(N-M)$ rate transition matrix Q, which is constructed from the states $\{M+1,\ldots,N\}$ of the BDP making the state M absorbing. Let the unique maximal real part of the eigenvalue be denoted by $\xi, \xi > 0$, then $r_{OL} = \xi$.

The parameters λ_{OL} (λ_{UL}) are taken as the sum of emission rates of all overload (underload) states, whereas the parameter r_{UL} is chosen in order to obtain a mean offered load ρ being equal to the mean emission rate of the BDP, namely $Np\lambda$, with the activity factor $p = \alpha/(\alpha+\beta)$. These three parameters are given by

$$\lambda_{OL} = \sum_{i=M+1}^{N} i\lambda\frac{\pi(i)}{\pi_{OL}} \tag{3.16}$$

$$\lambda_{UL} = \sum_{i=0}^{M} i\lambda\frac{\pi(i)}{\pi_{UL}} \tag{3.17}$$

$$r_{UL} = r_{OL}\frac{Np\lambda - \lambda_{UL}}{\lambda_{OL} - Np\lambda} \tag{3.18}$$

with $\pi_{OL} = \sum_{i=M+1}^{N}\pi(i)$ and $\pi_{UL} = \sum_{i=0}^{M}\pi(i)$; $\pi(i)$ is the limiting state probability of the BDP being in the state i, which is simply given by a binomial distribution with parameters $p = \alpha/(\alpha+\beta)$ and $q = \beta/(\alpha+\beta)$.

Although the problem in calculating factorial values for large N from a binomial distribution can be resolved by using Stirling's approximations [Feller, 1950], given as

$$n! \approx \sqrt{(2\pi n)}\, n^n \exp^{-n}, \tag{3.19}$$

this may introduce errors. Therefore we choose to calculate the distribution directly using the following recursive algorithm, starting with $\pi(0) = (1-p)^N$, where $\pi(i)$ is given by

$$\pi(i) \;=\; \pi(i-1)\frac{p}{q}\frac{N-i+1}{i} \tag{3.20}$$

3.3.3 WS Method

The method, proposed by Wang and Silvester [1993], differs from the BBLRW method in the determination of r_{OL}. Instead of approximating the distribution of the time spent in the overload region by an exponential distribution, the WS method uses the expected value of the time spent in the overload region, which is given by

$$\tau_{OL} \;=\; \frac{1}{N\beta\begin{pmatrix} N-1 \\ N-M-1\end{pmatrix}}\sum_{i=0}^{N-M-1}\begin{pmatrix} N \\ i\end{pmatrix}\left(\frac{\alpha}{\beta}\right)^{N-M-1-i} \tag{3.21}$$

This expected value is upper bounded by

$$\tau_{OL} \approx \frac{N-M}{(M+1)\beta} \tag{3.22}$$

Using (3.22), $r_{OL} = 1/\tau_{OL}$ can be found. After finding r_{OL}, the other three parameters of the MMPP model can be obtained in the same way as in the BBLRW method through (3.16)-(3.18). The use of (3.22) reduces the need for calculating a dominant eigenvalue, which can slow down the matching process especially for large N, as in the BBLRW method.

To extend the matching procedures to heterogeneous sources, an independent superposition of N_1 sources of class 1 and N_2 sources of class 2 will be considered. Let the pair (i, j) denote i sources of class 1 and j sources of class 2 which are in the ON state, then the state transition rate diagram for the aggregate source process can be as shown in Figure 3.5.

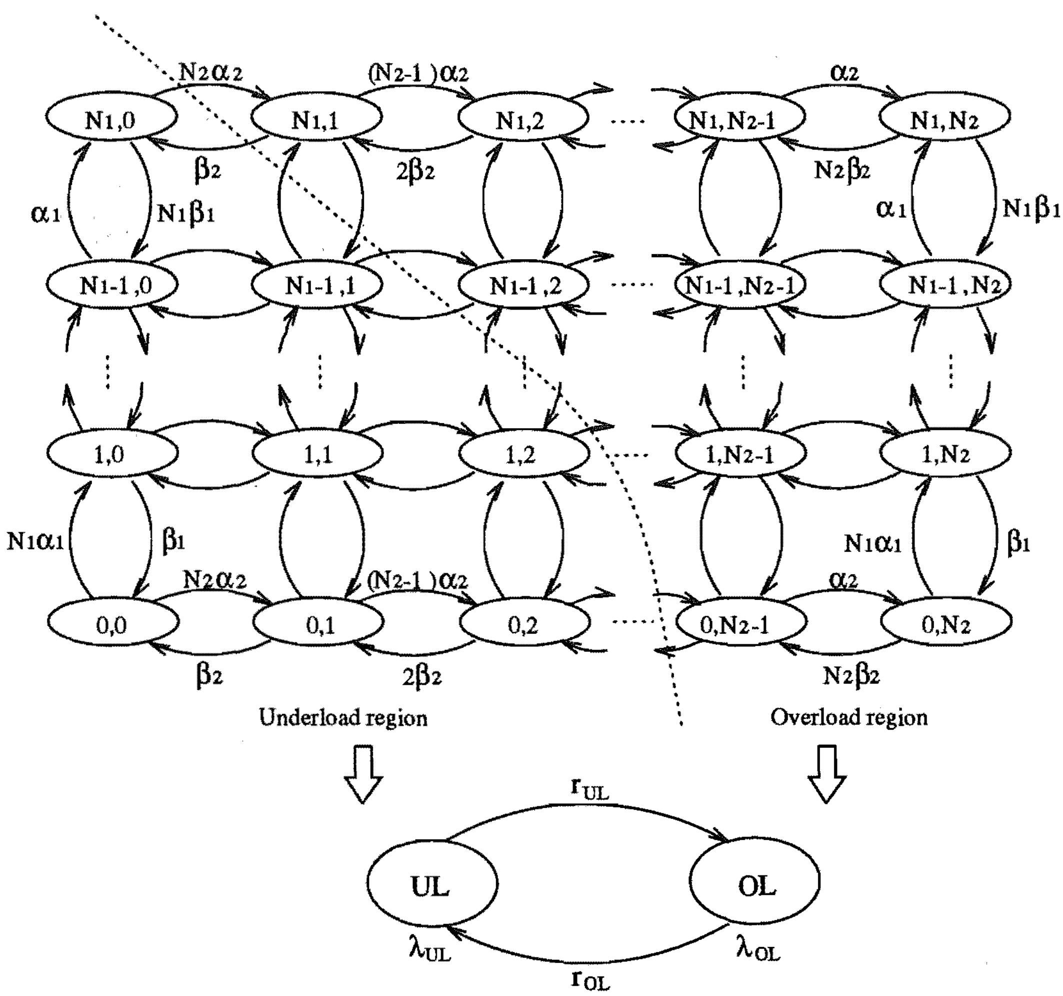


Figure 3.5 State transition diagram for heterogeneous superposition of IPP sources and its two-state MMPP approximation.

As for homogeneous sources, we can divide the state space into two regions, namely the overload and the underload regions as shown in Figure 3.5. We assume that the sum of peak arrival rate of N_1 sources is less than C, so that only class 2 sources will cause the overload. Then the maximum number of class 2 sources before an overload occurs is dependent on the number of ON sources from class 1 and it is given by

$$M_2(i) = \lfloor (C - i\lambda_1)/\lambda_2 \rfloor \tag{3.23}$$

with $i = 0, \ldots, N_1$. Following the same observation as for homogeneous sources, we can find that the MMPP parameters can be given as

$$r_{UL} = \frac{\sum_{i=0}^{N_1} M_2(i) + 1}{\sum_{i=0}^{N_1} N_2 - M_2(i)}(\beta_1 + \beta_2) \tag{3.24}$$

$$\lambda_{OL} = \sum_{i=0}^{N_1} \sum_{j=M_2(i)+1}^{N_2} (i\lambda_1 + j\lambda_2)\frac{\pi(i,j)}{\pi_{OL}} \tag{3.25}$$

$$\lambda_{UL} = \sum_{i=0}^{N_1} \sum_{j=0}^{M_2(i)} (i\lambda_1 + j\lambda_2)\frac{\pi(i,j)}{\pi_{UL}} \tag{3.26}$$

$$r_{UL} = r_{OL}\frac{(N_1 p_1 \lambda_1 + N_2 p_2 \lambda_2) - \lambda_{UL}}{\lambda_{OL} - (N_1 p_1 \lambda_1 + N_2 p_2 \lambda_2)} \tag{3.27}$$

Here $\pi(i,j)$ is the limiting state distribution given by the product of the limiting probabilities of the process being in state i (for $i = 0, \ldots, N_1$) and state j (for $j = 0, \ldots, N_2$), due to the independence assumption, i.e.

$$\pi(i,j) = \begin{pmatrix} N_1 \\ i \end{pmatrix} \begin{pmatrix} N_2 \\ j \end{pmatrix} \left(\frac{\alpha_1}{\alpha_1 + \beta_1}\right)^i \left(\frac{\beta_1}{\alpha_1 + \beta_1}\right)^{N_1 - i} \left(\frac{\alpha_2}{\alpha_2 + \beta_2}\right)^j \left(\frac{\beta_2}{\alpha_2 + \beta_2}\right)^{N_2 - j} \tag{3.28}$$

and

$$\pi_{OL} = \sum_{i=0}^{N_1} \sum_{j=M_2(i)+1}^{N_2} \pi(i,j), \quad \pi_{UL} = \sum_{i=0}^{N_1} \sum_{0}^{j=M_2(i)} \pi(i,j) \tag{3.29}$$

3.3.4 Numerical Results

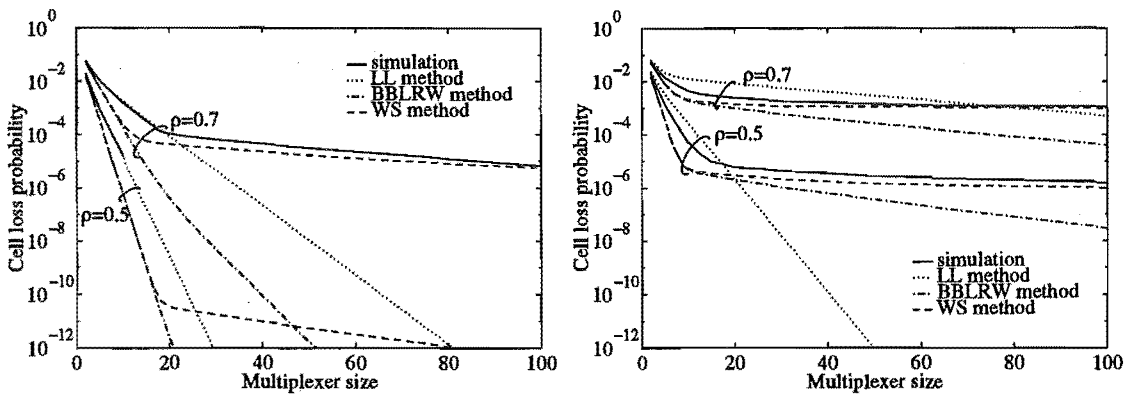
We consider two cases of homogeneous traffic, namely: a superposition of N voice sources feeding a $T1$ line with the capacity (C) approximately equal 1.536 Mbps [Heffes and Lucantoni, 1986; Sriram and Whitt, 1986], and a superposition of N video sources feeding an ATM link with C equal to 135.85 Mbps[1] [Maglaris *et al.*, 1988]. We assume an ATM cell payload (l_{ATM}) of 48 bytes. The parameters of the sources are listed in Table 3.1.

In comparing the performance of the matching procedures, we evaluated the cell loss probability for a finite queue fed by the resulting MMPP model using the MMPP/D/1/K analysis presented in Section 3.4. As a reference, we simulated the finite queue fed by N separate IPP sources and obtained its cell loss probability. To ensure the validity of the results, we require each point of the simulation results to have a 0.05 precision at a 95% confidence level. Time constraints on the simulation, without applying special techniques for simulating rare events, allowed us to obtain accurate results down to cell loss probabilities of 1×10^{-5} only.

[1] $C = 135.85$ Mbps is the assumed useful capacity of an ATM link of 155.52 Mbps [Baiocchi *et al.*, 1991].

Traffic sources	$\alpha\ (s^{-1})$	$\beta\ (s^{-1})$	λ(cells/s)
Voice	1.54	2.84	83.33
Video	0.64	3.25	11.10×10^3

Table 3.1 Traffic parameters.



(a) voice sources

(b) video sources

Figure 3.6 Homogeneous superposition of IPP sources.

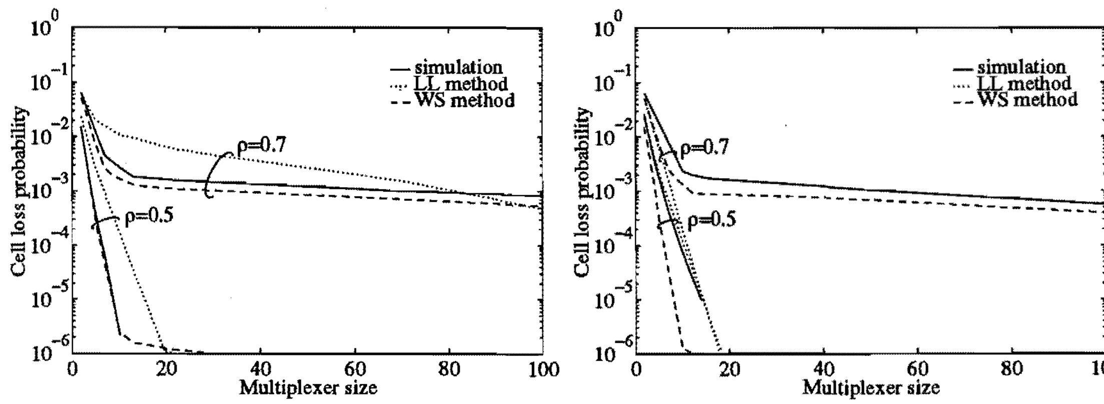
Figure 3.6 displays the cell loss probability, for both voice and video sources, as a function of multiplexer size with mean offered loads (ρ) of 0.5 and 0.7. From the figures, we immediately notice that the simulation curves are characterised by two distinct operation regions manifested by two different slopes. The higher slope is due to the faster decrease in *cell level congestion* with increasing buffer size, while the slower slope describes the slower *burst level congestion* decrease with increasing buffer size. This phenomena, referred to as the *kneeing effect*, has been observed by various authors [Baiocchi *et al.*, 1991; Norros *et al.*, 1991; Rasmussen *et al.*, 1991; Bonomi *et al.*, 1993]. The effect can be explained as follows.

We assume that the sum of the instantaneous bit rates of the sources exceeds the output link capacity. With a small buffer size, only few cells of the arriving bursts are accepted into the buffer. The time required to serve these cells is short and hence by the time the next bursts arrive, there is great chance that all cells have been served and the buffer is empty. This means that the actual load offered to the link bandwidth is less than 1.0 and the cell loss, characterising a cell level congestion, is due to the buffer size only. Increasing the buffer size reduces the loss. However, this also means that more cells will be accepted into the buffers. Since the link bandwidth is limited, the chance of emptying the buffer before the next bursts arrive is small. Increasing the buffer size further will not affect the results much, because the source of the problem is not in the buffer size anymore, but in the limited output link to serve the cells in the buffer. This explains the levelling off of the cell loss probability, which characterises the burst level congestion.

Comparing the results from the three matching methods to the simulation results, we can see that the LL method has a single slope which shows the cell level congestion phenomenon only.

It does not show any kneeing effect for both voice and video sources, and thus fails to model the overload in the approximate model. The method provides an extremely good matching for voice sources for low multiplexer size value ($K \leq 20$), but gives less accurate results for video sources. The BBLRW method demonstrates the kneeing effect for the video sources only, but perform badly for the voice sources, which are less bursty than the data traffic generated by sources assumed in [Baiocchi *et al.*, 1991] for demonstrating the performance of the method. This is due to the use of exponential distribution in approximating the distribution for the overload period. This approximation tends to underestimate the actual average of the time spent in the overload period. By avoiding such approximation and focusing on matching the actual number of states and the average time spent in each of the overload states, the WS method performs best among the three methods studied here. For varying multiplexer size, it performs best for smaller multiplexer size and worst at the kneeing point. As the number of sources (N), and thus the offered load increases, the approximation given by (3.22) is more justified and hence the accuracy of the results improves.

Proceeding on to a heterogeneous traffic condition, where a superposition of voice and video sources is involved, we consider two cases of the superposition. In the first case, we use 500 voice sources, while in the second case we use 1000 voice sources. We assume that the traffic generated by video sources is the main cause of system overload. For a given ρ, the number of video sources is equal to $N_2 = \lfloor(\rho C - N_1 p_1 \lambda_1)/(p_2 \lambda_2)\rfloor$. For example, at $\rho = 0.7$ and given $p_1 = 0.352$ and $p_2 = 0.165$ for the voice and video traffic parameters listed in Table 3.1, we can find $N_2 = 177$ for $N_1 = 500$ and $N_2 = 127$ for $N_1 = 1000$.



(a) 500 voice sources

(b) 1000 voice sources

Figure 3.7 Heterogeneous superposition of IPP sources.

Figure 3.7 displays the cell loss probability as a function of buffer size for both cases. The simulation results show that the effect of doubling the number of voice sources has insignificant effect on the performance of the system, which is dominated by the overload from video sources. This trend is demonstrated very well by the WS method while the LL method fails to capture this trend.

As a conclusion, the LL method, which has been shown to perform well on the basis of average cell delay [Lee and Lee, 1992], performs poorly on the basis of cell loss probability due to its inability to account for overload in the actual system. The BBLRW method, on the other hand, shows better capability to handle the overload in the system. However, the method is less accurate for less bursty sources, such as voice sources. A modification of the method, the WS method, has shown a much better agreement to the simulation results both in homogeneous as well as heterogeneous traffic conditions, and therefore it will be used throughout the rest of this chapter.

3.4 Multiplexer Analysis

Performance of ATM multiplexer has been widely studied by using matrix geometric approach [Neuts, 1981; Hou and Wong, 1990; Le Boudec, 1991], fluid approximation [Anick *et al.*, 1982; Elwalid and Mitra, 1992; Baiocchi *et al.*, 1992; Zhang, 1993] and some other innovative techniques [Louvion *et al.*, 1988; Norros *et al.*, 1991; Addie and Zukerman, 1993a]. Adopting a discrete-time model, we present an alternative solution to priority queueing problem by using an iterative computation method. A similar approach has been used previously in [Konheim, 1975; Tran-Gia, 1989; Lee and Lee, 1992; Bonomi *et al.*, 1993] for analysing non-priority queueing problems. It involves formulating the state probabilities at the end of a time slot, in terms of their values at the previous such time instant. This formulation can be written in term of convolution operation, which opens to a possibility of employing an efficient discrete transform algorithms, e.g. fast Fourier transform (FFT) [Henrici, 1979; Tran-Gia, 1989], in order to reduce computation costs. Furthermore due to its iterative nature, the solution can be made as accurate as possible to the actual solution by adjusting the stopping criterion.

3.4.1 System Assumptions

We assume a discrete-time system to capture the slotted nature of the ATM system, i.e. the time axis is divided into equal intervals of unit length Δ, called *slots*. Each slot is equal to the time interval required for transmitting a cell on the output link. Integer number $k \in \{1,2,3,\ldots\}$ are assigned to individual slot boundaries. The time interval $(k\Delta,(k+1)\Delta]$ will be referred to as the kth slot. Moreover, let us assume that

(A3.1) The multiplexer uses partial buffer sharing scheme [Kroner, 1990; Le Boudec, 1991], where in a buffer of size K, low priority cells are accepted only if the instantaneous buffer queue length at the cell arrival epoch is below a given threshold K_l $(K_l < K)$.

(A3.2) New cells are admitted into the multiplexer at the beginning of a slot.

(A3.3) Cell departure takes place at the end of a slot.

(A3.4) The multiplexer is in statistical equilibrium, which is reached when $k \to \infty$ and

$$\rho \triangleq \frac{\lambda_1 r_2 + \lambda_2 r_1}{r_1 + r_2} < 1 \tag{3.30}$$

with $\lambda_1, \lambda_2, r_1$, and r_2 being the four parameters of the MMPP model (see Figure 3.3).

The multiplexer performance is measured by the cell loss probability for high and low priority cells.

3.4.2 Mathematical Preliminaries and Notations

We define an iteration cycle to be the time from the beginning of the first slot in State 1 to the end of the last slot in State 2 of an MMPP model. In the analysis, the following symbols will be used and a sample path of the state process that shows the relationship between these variables are depicted in Figure 3.8

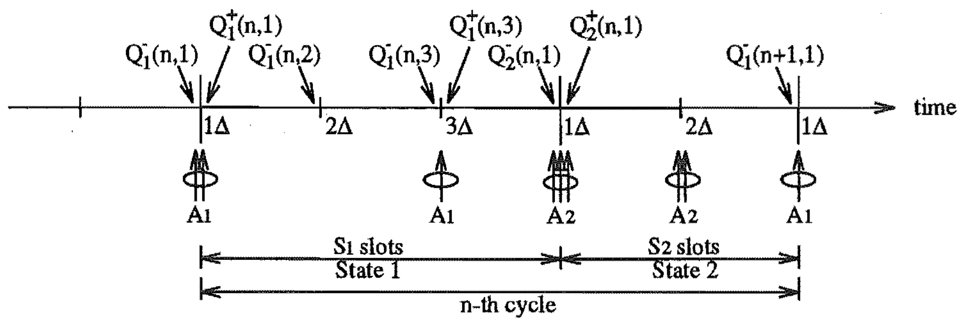


Figure 3.8 A sample path for random variables.

S_1, S_2	the sojourn times in State 1 and State 2.
A_1, A_2	the batch size of cells arriving during a slot time when the MMPP source is in State 1 and State 2, respectively.
$Q_1^-(n,k), Q_1^+(n,k)$	the queue lengths, respectively, immediately prior to and immediately after the beginning of the kth slot in State 1 of the nth cycle.
$Q_2^-(n,k), Q_2^+(n,k)$	the queue lengths, respectively, immediately prior to and immediately after the beginning of the kth slot in State 2 of the nth cycle.
K_l, K	the buffer thresholds for low and high loss priority cells, respectively.

We define the following Σ operators [Tran-Gia, 1986; Murata *et al.*, 1990]

$$\Sigma^m(y(j)) = \begin{cases} y(j) & j < m \\ \sum_{i=m}^{\infty} y(i) & j = m \\ 0 & j > m \end{cases} \tag{3.31}$$

$$\Sigma_0(y(j)) = \begin{cases} 0 & j < 0 \\ \sum_{i=-\infty}^{0} y(i) & j = 0 \\ y(j) & j > 0 \end{cases} \tag{3.32}$$

and the discrete convolution operator $*$ as

$$y(j) = y_1(j) * y_2(j) \triangleq \sum_{i=-\infty}^{\infty} y_1(j-i)y_2(i) \tag{3.33}$$

Refering to the sample path in Figure 3.8, we notice that a cycle consists of S_1 slots in State 1 and S_2 slots in State 2. The probability mass functions $\{s_1(k)\}, k = 1,2,\ldots$ of S_1, and $\{s_2(k)\}$ of S_2 are geometrically distributed, i.e.

$$s_1(k) = r_1\,(1-r_1)^{k-1} \tag{3.34}$$

$$s_2(k) = r_2\,(1-r_2)^{k-1} \tag{3.35}$$

Considering an individual slot k in State 1, the queue lengths immediately prior to and immediately after the beginning of the time slot is related by

$$Q_1^+(n,k) \;=\; \min(Q_1^-(n,k) + A_1, K+1) \tag{3.36}$$

At the end of the slot the cell in service, if any, will depart. Thus, we have

$$Q_1^-(n,k+1) \;=\; \max(Q_1^+(n,k) - 1, 0) \tag{3.37}$$

Let define $\{q_1^-(n,k,j)\}$ and $\{q_1^+(n,k,j)\}, j = 0,1,\ldots,K$, be the probability mass functions for $Q_1^-(n,k)$ and $Q_1^+(n,k)$, respectively, and $\{a_1(j)\}, j = 0,1,\ldots,$ be the probability mass function of A_1. Using the operators in (3.31)-(3.33), we can write (3.36) and (3.37) as

$$\begin{array}{rcll} q_1^+(n,k,j) & = & \Sigma^{K+1}(q_1^-(n,k,j) * a_1(j)) & \text{if } 0 \le j \le K+1 \\ q_1^-(n,k+1,j) & = & \Sigma_0(q_1^+(n,k,j+1)) & \text{if } 0 \le j \le K \end{array} \tag{3.38}$$

The equations (3.36)-(3.38) are also valid for State 2, except with the change of index 1 to 2.

In order to relate the queue length for State 2 to the queue length for State 1 in the nth cycle, we observe from Figure 3.8 that the queue length $Q_2^-(n,1)$, immediately prior to the beginning of the first slot in State 2, is related to the queue length $Q_1^-(n, S_1+1)$, where S_1 is the length of the sojourn time at State 1. If the sojourn time is of fixed length T, we will expect that $Q_2^-(n,1) = Q_1^-(n,T+1)$ with a unity probability. However, since the length is geometrically distributed, we need to weigh $Q_1^-(n,S_1+1)$ with the probability mass function $\{s_1(k)\}$ of S_1, given by (3.34). This relationship can be expressed as

$$q_2^-(n,1,j) \;=\; \sum_{k=1}^{\infty} s_1(k)\, q_1^-(n,k+1,j) \tag{3.39}$$

Similarly, $Q_1^-(n+1,1)$ is related to $Q_2^-(n,k)$ by

$$q_1^-(n+1,1,j) \;=\; \sum_{k=1}^{\infty} s_2(k)\, q_2^-(n,k+1,j) \tag{3.40}$$

Under steady state conditions, we have

$$q_1^-(k,j) \;=\; \lim_{n\to\infty} q_1^-(n,k,j) \tag{3.41}$$

This steady state probability can be obtained by iterating the process until the desired accuracy, determined as

$$\left|\frac{q_1^-(n,j) - q_1^-(n-1,j)}{q_1^-(n,j)}\right| < \epsilon \tag{3.42}$$

is reached, where $q_1^-(n,j) = \sum_{k=1}^{\infty} q_1^-(n,k,j)$ and $\epsilon = 1\times 10^{-6}$. After finding $q_1^-(k,j)$ and $q_2^-(k,j)$, we can determine the cell loss probability as will be discussed in the next sections.

3.4.3 MMPP Source without Pretagged Cells (Pure Traffic)

In this section, we will derive the cell loss probability based on the formula for the long term averages

$$cell\ loss\ probability = \lim_{t\to\infty} \frac{number\ of\ cells\ lost\ during\ interval\ t}{offered\ load\ during\ interval\ t} \tag{3.43}$$

In deriving the average number of cells lost, we first derive the average cell loss per slot in State $1(L_1)$ by observing an arbitrary cell in the group arriving in a slot. The probability of the cell being in an arrival group of size i is $a_1(i)$, given as

$$a_1(i) = \lambda_1^i \exp(-\lambda_1)/i\,! \tag{3.44}$$

Since the system can hold up to K cells in the buffer in addition to the cell in service, hence blocking will occur if there are more than $(K - j + 1)$ cells in the arriving new batch when the queue length is j. This leads to an average cell loss in slot k,

$$L_1(k,j) = \sum_{i=K-j+2}^{\infty} [i - (K - j + 1)]\, a_1(i) \tag{3.45}$$

Unconditioning the queue length j and the slot k, we get

$$L_1 = \sum_{k=1}^{\infty} s_1(k) \sum_{j=0}^{K} q_1^-(k,j)\, L_1(k,j) \tag{3.46}$$

The equations (3.45) and (3.46) are also applicable for State 2 by changing the index. To obtain the overall number of cell loss for both states, we weighted the cells lost in each state by the probability of being in the state, resulting in

$$\begin{aligned} L &= \frac{r_1^{-1}}{r_1^{-1} + r_2^{-1}} L_1 + \frac{r_2^{-1}}{r_1^{-1} + r_2^{-1}} L_2 \\ &= \frac{r_2 L_1 + r_1 L_2}{r_1 + r_2} \end{aligned} \tag{3.47}$$

With an average number of cells arriving per slot being given as

$$\rho = \frac{r_2\lambda_1 + r_1\lambda_2}{r_1 + r_2} \tag{3.48}$$

we can find the average cell loss probability as

$$P = \frac{r_2 L_1 + r_1 L_2}{r_2\lambda_1 + r_1\lambda_2} \tag{3.49}$$

3.4.4 MMPP Source with Pretagged Cells (Mixed Traffic)

Let us consider an MMPP source with pretagged cells, where the source generates both high and low priority cells. Given that the fraction of high priority cells to total number of arriving cells is η

and the overall arriving rate is λ_1, the probability mass function $\{a_{1h}(i)\}$ and $\{a_{1l}(i)\}, i = 1, 2, \ldots$ for high and low priority traffic, respectively, can be given by

$$a_{1h} = \lambda_{1h}^i \exp(-\lambda_{1h})/i\,! \tag{3.50}$$

$$a_{1l} = \lambda_{1l}^i \exp(-\lambda_{1l})/i\,! \tag{3.51}$$

with $\lambda_{1h} = \eta\lambda_1$ and $\lambda_{1l} = (1-\eta)\lambda_1$.

Since the low priority cells can access the buffer up to K_l only, while the high priority cells can access the buffer up to K, we must condition the probability mass function of $\{a_1(i)\}$ in (3.38), denoted here as $\{a_1(i|j)\}$, by the queue length upon the cell arrivals, $Q_1^-(n,k) = j$, i.e.

$$a_1(i|j) = \begin{cases} a_1(i) & \text{if } 0 \le j \le K_l \\ a_{1h}(i) & \text{if } K_l + 1 \le j \le K \end{cases} \tag{3.52}$$

This condition is also applied when we derive the average number of cells lost for both high and low priority cells in each state.

As in the previous section, when the queue length is j, cell loss occurs if there are more than $(K - j + 1)$ cells arriving within a slot and the number of average cells lost $L_1(k,j)$ is given by (3.45). For $0 \le j \le K_l$, this cell loss comprises both the high and low priority cells. On average, the cell loss for each priority can be given as $L_1(k,j)$ weighted by the average number of arrivals of each priority cells, i.e.

$$L_{1h}(k,j) = \eta L_1(k,j) \tag{3.53}$$

$$L_{1l}(k,j) = (1-\eta)L_1(k,j) \tag{3.54}$$

For $K_l + 1 \le j \le K$, all low priority cells will be lost, which leaves the cell loss for the high priority dependent on the number of high priority arrivals. This results in the average number of lost cells being given as

$$L_{1h}(k,j) = \sum_{i=K-j+2}^{\infty} [i - (K - j + 1)]\, a_{1h}(i) \tag{3.55}$$

$$L_{1l}(k,j) = \sum_{i=1}^{\infty} i\, a_{1l}(i) \tag{3.56}$$

After finding $L_{1h}(k,j)$ and $L_{1l}(k,j)$ and knowing that the probability mass function $\{s_1(k)\}$ is given in (3.34), we can unconditioned them for j and k as in (3.46), i.e.

$$L_{1h} = \sum_{k=1}^{\infty} s_1(k) \sum_{j=0}^{K} q_1^-(k,j)\, L_{1h}(k,j) \tag{3.57}$$

$$L_{1l} = \sum_{k=1}^{\infty} s_1(k) \sum_{j=0}^{K} q_1^-(k,j)\, L_{1l}(k,j) \tag{3.58}$$

and finally arrive at the average cell loss probability as

$$P_h = \frac{r_2 L_{1h} + r_1 L_{2h}}{r_2\lambda_{1h} + r_1\lambda_{2h}} \tag{3.59}$$

$$P_l = \frac{r_2 L_{1l} + r_1 L_{2l}}{r_2\lambda_{1l} + r_1\lambda_{2l}} \tag{3.60}$$

3.4.5 Numerical Results

In this section we present simulation results to verify the analysis and obtain numerical examples. All simulation results represent steady state values with the relative precision below 0.05 at 95% confidence level. Using the video source in Section 3.3.4 as an example, we can find the four parameters of the MMPP model for $\rho = 0.7$ and $\rho = 0.9$, normalised to the output line capacity (C) of 135.85 Mbps, by using the WS method as given in Table 3.2.

	λ_1	λ_2	r_1	r_2
$\rho = 0.7$	0.690	1.044	5.429×10^{-8}	2.827×10^{-6}
$\rho = 0.9$	0.827	1.088	7.739×10^{-7}	2.056×10^{-6}

Table 3.2 MMPP parameters.

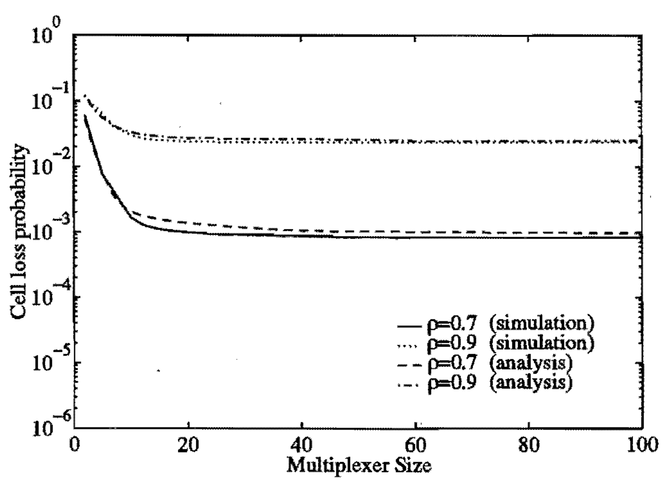
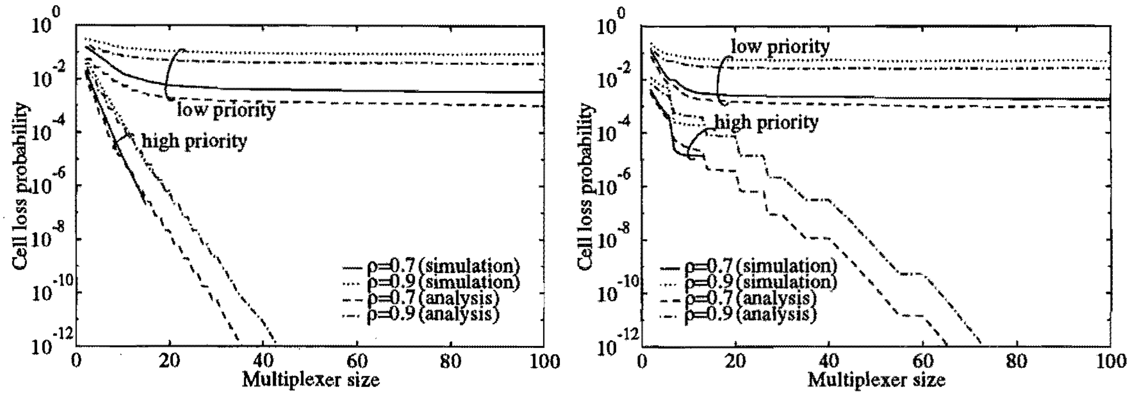


Figure 3.9 Loss probability versus multiplexer size for non-priority system.

We firstly simulate a non-priority multiplexer system fed by the MMPP model. Figure 3.9 shows the computed cell loss probability as a function of the multiplexer sizes and the values estimated by simulation. The figure shows a good agreement between analytical and simulation results for whole range of multiplexer sizes and for varying mean offered load.

For verifying the analysis of a priority multiplexer system, we assume a fraction of high priority traffic η to be 0.15 and 0.5, which represents typical lower and upper bounds of ATM traffic sources [Sriram *et al.*, 1991; Kroner *et al.*, 1991]. The ratio of the low priority buffer threshold K_l to the overall buffer size, denoted as $\kappa = K_l/K$, is set at 0.85 and 0.5, for $\eta = 0.15$ and $\eta = 0.5$, respectively.

Figure 3.10(a) shows the computed cell loss probability as a function of the multiplexer sizes and the values obtained from simulation. The figure shows a reasonable agreement (at 95% confidence level) of the loss probabilities for high priority cells between the analytical and the simulation results. The ripple in the results, especially for small buffer size, is due to the rounding



(a) $\eta = 0.5, \kappa = 0.5$ (b) $\eta = 0.15, \kappa = 0.85$

Figure 3.10 Loss probability versus multiplexer size for priority system.

of the threshold values, i.e. $\lfloor \kappa K \rfloor$. This ripple demonstrates the sensitivity of high priority cells to the threshold values. On the other hand, the loss probabilities for low priority cells are less sensitive to the threshold values, but strongly affected by the overload in the system. The discrepancies in the analytical results are due to the assumption that low priority cells can still be accepted during the transition from states $j \leq K_l$ to states $j > K_l$. We expect that with smaller η and higher κ values, as demonstrated in Figure 3.10(b), these discrepancies will be minimised. In the figure, we can also notice the strong rippling effect for the loss probability for high priority cells. This confirms our initial observation about the sensitivity of high priority cells to the size of the buffer threshold. A further confirmation can be observed by plotting the cell loss probability, obtained from analytical solutions, against the fraction of high priority traffic η for different buffer threshold ratio κ, as shown in Figure 3.11.

The figure shows that variations of η and κ affect only the loss probability for high priority traffic, with no much affect on the low priority traffic. This is due to the overloaded nature in the system. And again the plot shows the weaker sensitivity of the low priority cells to the changing of the fraction of high priority traffic or the buffer threshold ratio. This observation is different from the conclusion drawn by Meyer *et al.* [1993] for Poisson input traffic that changing either the fraction of high priority traffic or the buffer threshold ratio affects both traffic priorities. Overall these results indicate that the cell loss priority is expected to play a more significant role in a bursty traffic environment and highlight the importance of choosing the right buffer threshold values. For doing this, the importance relationship depicted in Figure 3.12 can be helpful.

Figure 3.12 demonstrates that for a given offered load ρ and fraction of high priority traffic η, the difference in the order of magnitude of the loss probabilities of low and high priority cells ($\log_{10}(P_l) - \log_{10}(P_h)$) will remain constant for varying buffer size, provided that the difference in the buffer thresholds ($K - K_l$) is fixed. This relationship has been previously observed in [Meyer *et al.*, 1993; Kroner *et al.*, 1991], for Poisson traffic input. However, our results again highlight the effects of bursty traffic which produces small differences between the results for $\rho = 0.4$ and $\rho = 0.7$, but a strong dependency on the traffic mixture η. Weaker dependency on ρ is beneficial in developing a bandwidth allocation method, as will be discussed in Section 3.5.1.

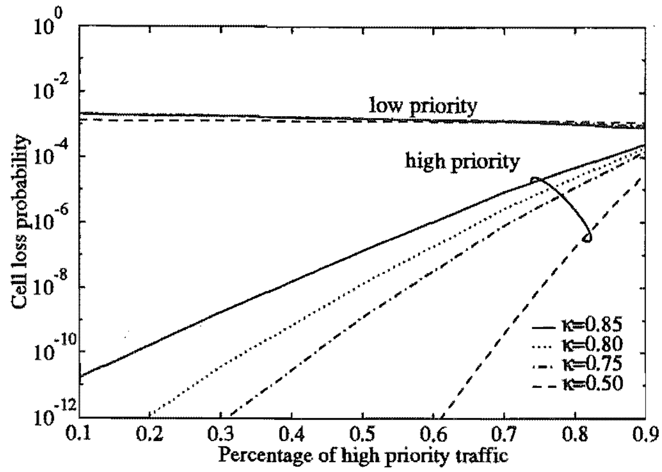
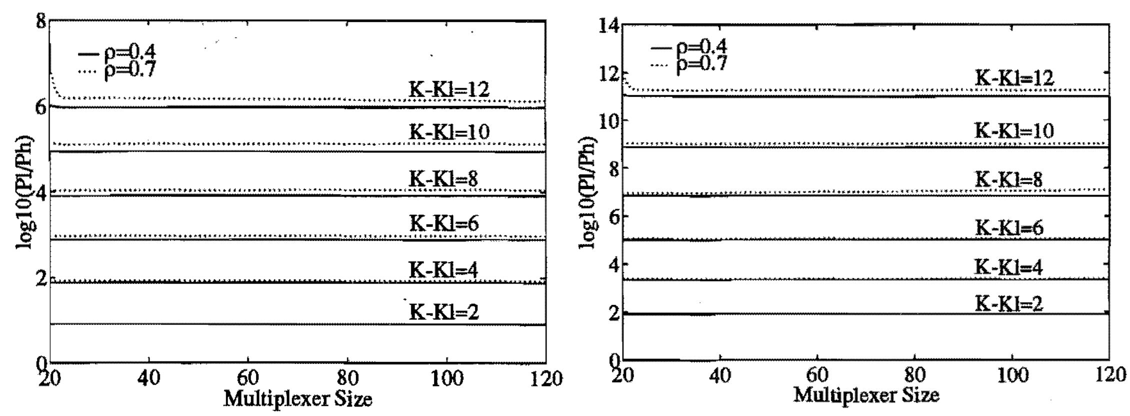


Figure 3.11 Loss probability versus high priority load ratio ($K = 50, \rho = 0.7$).



(a) $\eta = 0.5, \kappa = 0.5$

(b) $\eta = 0.15, \kappa = 0.85$

Figure 3.12 Difference between the loss probabilities of high and low priority cells ($\rho = 0.7$).

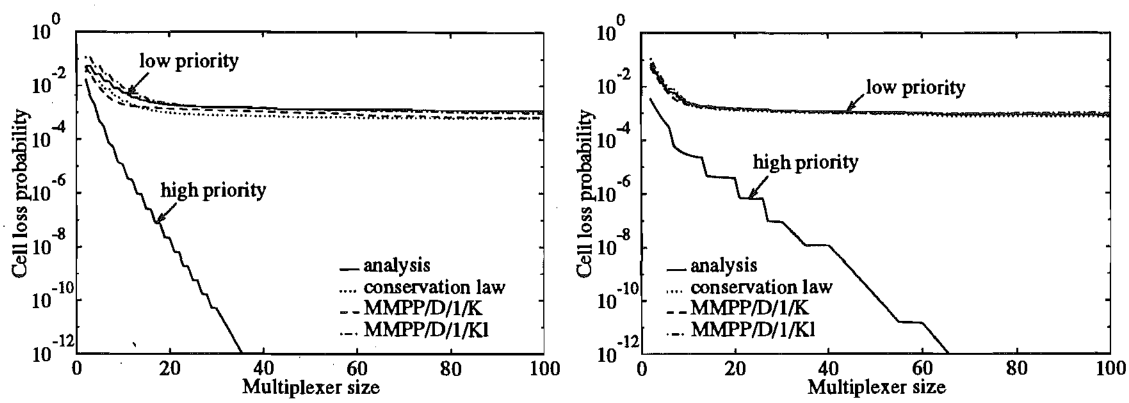
Another important relationship in the priority multiplexing system is commonly referred to as the *conservation law*, which has been formulated by Sumita and Ozawa [1988] for the *push-out* scheme with Poisson traffic input. The law states that

> *the loss probabilities P_h and P_l for high and low priority cells in a push-out scheme $M_h + M_l/G/1/K$ are related to the loss probability P of ordinary M/G/1/K queueing system with aggregate arrival rate $\lambda = \lambda_h + \lambda_l$,* namely

$$\lambda P = \lambda_h P_h + \lambda_l P_l \tag{3.61}$$

Restating the conservation law in terms of η for the case of MMPP input traffic, we have

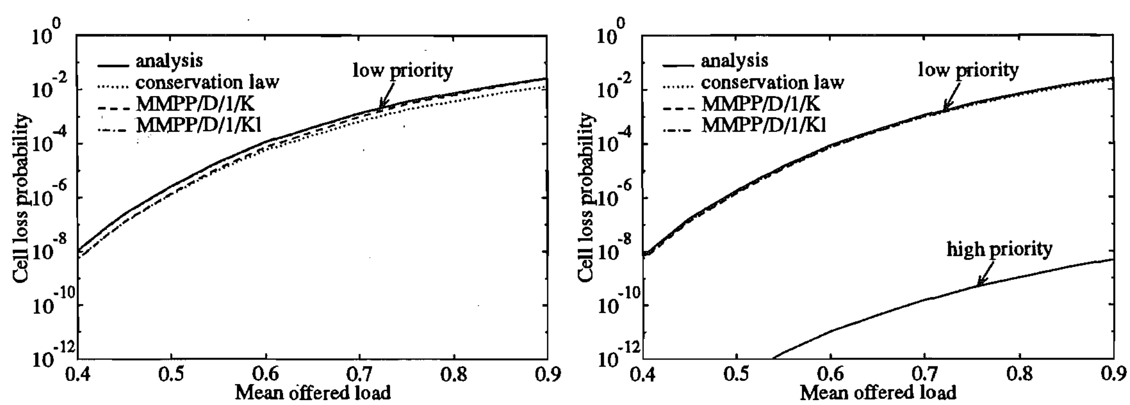
$$P = \eta P_h + (1 - \eta) P_l \tag{3.62}$$



(a) $\eta = 0.5, \kappa = 0.5$

(b) $\eta = 0.15, \kappa = 0.85$

Figure 3.13 Conservation law for varying multiplexer size ($\rho = 0.7$).



(a) $\eta = 0.5, \kappa = 0.5$

(b) $\eta = 0.15, \kappa = 0.85$

Figure 3.14 Conservation law for varying offered load ($K = 50$).

In Figures 3.13 and 3.14, we plot the loss probabilitites for both the high and low priority cells and the equivalent loss probabilities obtained from the conservation law for varying multiplexer sizes and varying offered traffic, respectively. From the figures, we can notice the close agreement between the results from the conservation law and the actual loss probabilities from MMPP/D/1/K analysis. The slight discrepancies in the results for $\eta = 0.5$ which show that the loss probabilities from the conservation law are smaller than the ones from MMPP/D/1/K are due to the discrepancies in the multiplexer analysis as discussed early in this section.

Unlike in the push-out scheme where both high and low priority cells can access the buffer up to K, in the multiplexer of our consideration the low priority cells can access the buffer only up to $K_l (K_l < K)$. This means that it is possible that at any stage, part of the buffer may be unused whereas low priority cells are discarded. Therefore the conservation law does not provide the exact relationship. For comparison purposes, we also plot the loss probabilities for $MMPP/D/1/K_l$, which can be considered as the upper bound to the conservation law. The results, however, do not show any significant differences.

3.5 Bandwidth Allocation Algorithms

With the existence of low priority cells in mixed connections, we restate the bandwidth allocation problem found in [Guerin *et al.*, 1991; Monteiro *et al.*, 1991] as follows:

> *Given a mix of N sources making use of cell loss priority and sharing a transmission link with buffer size K, estimate the link bandwidth W and the buffer threshold K_l that is required to satisfy QoS requirements for both high and low priority cells.*

In the following sections, we discuss some solutions to the problem, considering both homogeneous and heterogenous traffic environments.

3.5.1 Homogeneous Traffic Environment

In this section, we investigate six alternatives for allocating the required bandwidth for homogeneous connections (i.e. the traffic within all connections has the same characteristics). The first four methods can be found in the literatures, while the last two, Methods V and VI, are new methods proposed in this thesis.

Method I

This method is commonly used for pure connections, for example, it can be found in [Lee and Lee, 1992]. For mixed connections, the method would have to assume that the traffic comprises single priority cells only and allocates an effective bandwidth for satisfying the most stringent QoS requirements, namely the QoS of high priority traffic. Prior to evaluating the effective bandwidth, a buffer size K must be chosen to meet the most stringent cell delay requirement under a FIFO discipline without introducing any time priority, that is

$$K = \left\lceil \frac{Ct_{max}}{l_{ATM}} \right\rceil \tag{3.63}$$

where C is the output line capacity, t_{max} is the maximum allowable cell delay at the buffer and l_{ATM} is the number of bits per ATM cell. Normally a value of $K = 50$ is sufficient to keep average cell delay below $100\mu s$ when $C = 135.85 Mbps$.

After determining the buffer size, a *bisection* algorithm is used to search for the effective bandwidth in the range $Np\lambda < W < N\lambda$. This algorithm is found to converge faster than the logarithmic interpolation algorithm used in [Gallassi *et al.*, 1990a; Monteiro *et al.*, 1991] for bursty traffic. It is given as follows.

Algorithm 3.1.

Step 1. Assign initial points: $x_1 = Np\lambda$ and $x_2 = N\lambda$.

Step 2. Form $x_3 = (x_1 + x_2)/2$ and match the statistical characteristic of the superposition to the MMPP parameters $(\lambda_1, \lambda_2, r_1, r_2)$ by using the WS method.

Step 3. Evaluate the cell loss probability y_3 for a given K and with the above MMPP parameters by using formulas given in Section 3.4.3.

Step 4. Substitute :
if $(y_3 > QoS)$ then do $x_1 \leftarrow x_3; y_1 \leftarrow y_3$ enddo.
else do $x_2 \leftarrow x_3; y_2 \leftarrow y_3$ enddo.

Step 5. Repeat steps 2-4 until $|(y_3 - QoS)/QoS| \leq \epsilon$, where ϵ is the required precision (default value $\epsilon = 10^{-6}$).

Step 6. $W = x_3$.

Method II

This method was initially proposed by Saito [1992] as a connection admission algorithm for two classes of virtual channels with different QoS requirements, while within the virtual channels all cells requires the same QoS. Here we consider the case where cells within the virtual channels have different QoS requirements.

Using the method, bandwidth will have to be assigned based on the peak bit rate of the high priority traffic only, in which case any assigned bandwidth unused by the high priority traffic can be utilised by the low priority traffic. This means that only the QoS of the high priority cells is guaranteed. Therefore, in order to prevent low priority cells from competing for the bandwidth with high priority cells, a buffer threshold K_l needs to be chosen such that the QoS for the high priority cells can still be satisfied irrespective the offered load from the low priority traffic. Further discussion on this issue can be found in Chapter 5.

Method III

This method is based on the class related rule (CRR), proposed by Gallassi *et al.* [1990b]. For a mixed connection, this method basically treats the high and low priority traffic as two separate classes of traffic. Given the amount of offered traffic ρ_h and ρ_l and the traffic parameters of the high and low priority classes, the assigned bandwidths, W_h and W_l, needed to satisfy the QoS requirements for high and low priority traffic, QoS_h and QoS_l, are found using the following algorithm.

Algorithm 3.3.

Step 1. Assign initial values: $x_1 = \eta N p \lambda$ and $x_2 = \eta N \lambda$.

Step 2. Obtain W_h by following steps 2-6 from Algorithm 3.1 with $QoS = QoS_h$.

Step 3. Assign new values: $x_1 = (1 - \eta) N p \lambda$ and $x_2 = (1 - \eta) N \lambda$.

Step 4. Obtain W_l by following steps 2-6 from Algorithm 3.1 with $QoS = QoS_l$.

Step 5. $W = W_h + W_l$.

The bandwidth required by the connection is simply the sum of the assigned bandwidths, W_h and W_l. In order to avoid allocating more bandwidth than Method I, an additional constraint is sometimes placed that the bandwidth allocated should be the minimum of the sum of W_h and W_l or the one determined by Method I (W_I), i.e.

$$W = \min(W_I, W_h + W_l) \tag{3.64}$$

Method IV

This method is a mathematical formulation of the equivalent bandwidth assignment curve proposed by Gallassi *et al.* [1990b]. It is similar to Method III, except that the total offered load, instead of the offered load of individual classes, is used in determining the required bandwidths W_h and W_l for two different QoS requirements, QoS_h and QoS_l. The bandwidth required by the connection is simply a linear combination of W_h and W_l, i.e.

$$W = \eta W_h + (1 - \eta) W_l \tag{3.65}$$

The method can be formulated as follows.

Algorithm 3.4.

Step 1. Assign initial values: $x_1 = Np\lambda$ and $x_2 = N\lambda$.

Step 2. Obtain W_h by following steps 2-6 from Algorithm 3.1 with $QoS = QoS_h$.

Step 3. Assign new values: $x_1 = Np\lambda$ and $x_2 = N\lambda$.

Step 4. Obtain W_l by following steps 2-6 from Algorithm 3.1 with $QoS = QoS_l$.

Step 5. $W = \eta W_h + (1 - \eta) W_l$.

The allocated bandwidth is the same as that of Method I with the QoS requirement being QoS_l when $\eta = 0$ and QoS_h when $\eta = 1.0$.

Method V

From Algorithms 3.3 and 3.4, we can notice that in determining the required bandwidth, Methods III and IV search for two assigned bandwidths corresponding to two different QoS requirements in mixed connections. This implies a need for executing Algorithm 3.1 twice and hence obviously the methods are more time consuming than Method I. In order to overcome this drawback, it is desirable to devise a method which only requires a single bandwidth search. Such a method obviously needs to rely on a relationship between QoS requirements for high and low priority traffic in such a way that if the QoS for high priority traffic is satisfied for a given assigned bandwidth, then the QoS for low priority traffic will also be satisfied through the relationship. One such relationship is provided by the conservation law, which was discussed in Section 3.4.5.

Utilising the conservation law, we propose a method which first determines the equivalent cell loss probability (QoS_e) based on (3.62) and then calculates an equivalent bandwidth using Algorithm 3.1 with the QoS requirement being QoS_e. The procedure can be formulated as

Algorithm 3.5.

Step 1. Determine the equivalent QoS requirement through the conservation law in (3.62), i.e. $QoS_e = \eta QoS_h + (1 - \eta) QoS_l$

Step 2. Assign initial values: $x_1 = Np\lambda$ and $x_2 = N\lambda$.

Step 3. Obtain W by following steps 2-6 from Algorithm 3.1 with $QoS = QoS_e$.

After finding the bandwidth to satisfy QoS_e, we can choose the buffer threshold K_l to just satisfy the QoS requirement for high priority traffic using a bisection method and the priority multiplexing analysis in Section 3.4.4. Based on the conservation law, we can expect that the QoS requirement for the low priority traffic will also be satisfied. The justification for using the conservation law in this method can be seen in Figure 3.16, where we plot the cell loss probabilities for high and low priority traffic from Method V.

Method VI

The Methods I-V considered so far have made use of analytical solutions for a non-priority multiplexing system in finding the required bandwidth. This results in the methods being unable to fully exploit the statistical multiplexing of high and low priority traffic and the presence of the additional buffer threshold which limits the low priority traffic. Method III, for example, does not take any advantage of the statistical multiplexing property as the low and high priority traffic have been considered separately.

In order to overcome this drawback, it is preferable to use analytical solutions for a priority multiplexing system to find the required bandwidth. However, there is a new problem associated with the approach, namely the need to choose the initial K_l value and to find a link bandwidth to satisfy the different QoS requirements simultaneously. Fortunately, the invariant property of the difference between the low and high loss probabilities $(log_{10}(P_l) - log_{10}(P_h))$ for a given offered load ρ, discussed in Section 3.4.5, offers a solution to the problem. The relationship allows us to find $K - K_l$, and hence the buffer threshold K_l, to meet the different QoS requirements. We can then find the required bandwidth in order to satisfy the QoS for high priority traffic by using the multiplexer analysis presented in Section 3.4.4. In meeting the QoS for high priority traffic, we will also satisfy the QoS for the low priority traffic.

Based on the relationship, we propose a new method for assigning bandwidths to mixed connections which can be defined as follows.

Algorithm 3.6.

Step 1. Calculate the difference in the QoS requirements, i.e. $(log_{10}(QoS_l) - log_{10}(QoS_h))$.

Step 2. Using the graph as in Figure 3.12, determine the value of $K - K_l$ and hence the buffer threshold K_l.

Step 3. Assign initial values: $x_1 = Np\lambda$ and $x_2 = N\lambda$.

Step 4. Form $x_3 = (x_1 + x_2)/2$ and match the statistical characteristic of the superposition to the MMPP parameters $(\lambda_1, \lambda_2, r_1, r_2)$ by using the WS method.

Step 5. Evaluate the cell loss probability y_3 for given K, K_l and with the above MMPP parameters by using analysis in Section 3.4.4.

Step 6. Substitute :

if $(y_3 > QoS_h)$ then do $x_1 \leftarrow x_3$; $y_1 \leftarrow y_3$ enddo.

else do $x_2 \leftarrow x_3$; $y_2 \leftarrow y_3$ enddo.

Step 7. Repeat steps 4-6 until $|(y_3 - QoS_h)/QoS_h| \leq \epsilon$, where ϵ is the required precision (default value $\epsilon = 10^{-6}$).

Step 8. $W = x_3$.

Performance Comparisons

For comparing the performance of the methods, we take the video sources used in Section 3.3.4 as an example. We choose $K = 50$, $QoS_l = 10^{-6}$, $QoS_h = 10^{-9}$ as references. Two different fractions of high priority traffic, $\eta = 0.15$ and 0.5, are considered. The buffer threshold K_l used in Method VI is chosen from the graph in Figure 3.12, where $K - K_l = 3$ for $\eta = 0.15$ and $K - K_l = 6$ for $\eta = 0.5$ are required to satisfy the three orders of magnitude difference between the high and low priority QoS.

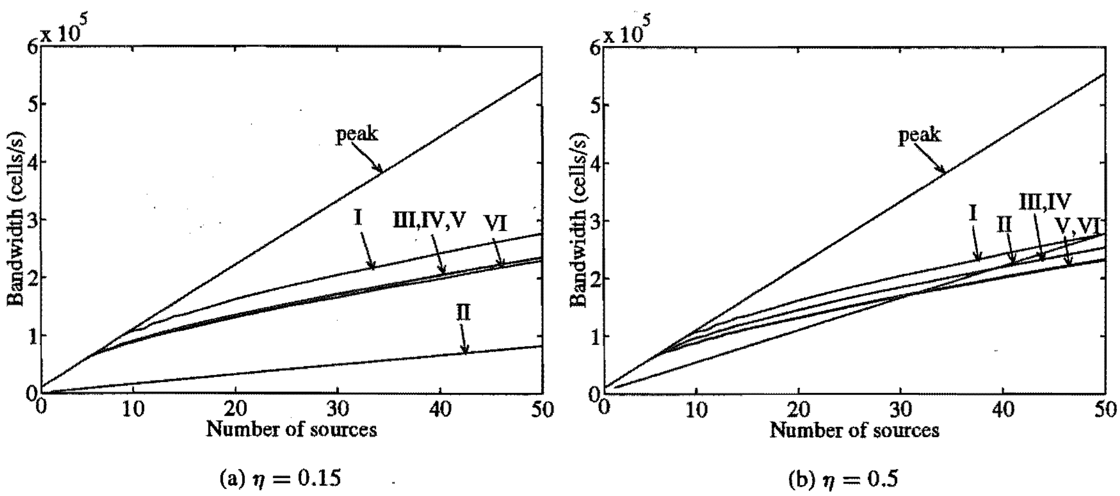
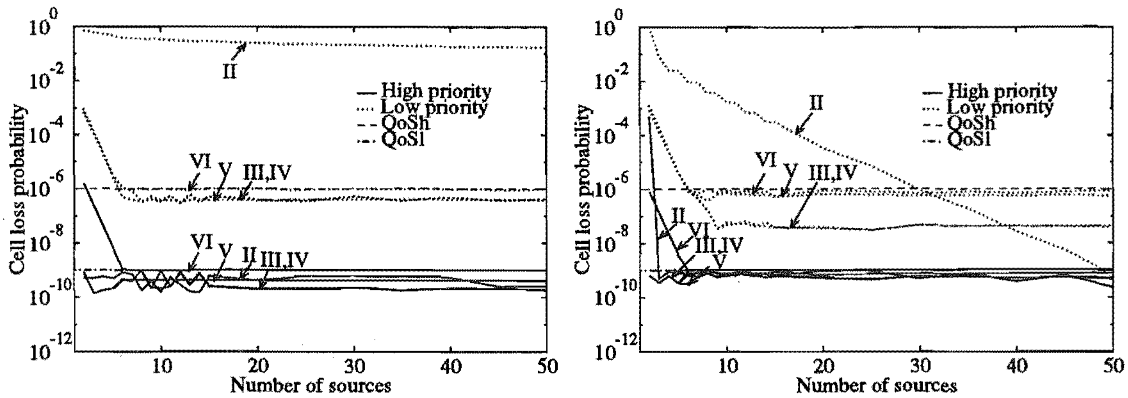


Figure 3.15 Assigned bandwidth versus the number of video sources for $QoS_l = 10^{-6}$ and $QoS_h = 10^{-9}$.

Figures 3.15 shows the bandwidth requirement against varying number of video sources (N) for the six methods and compares them with the method using peak bandwidth assignment. From the figure, we first notice that when more sources are multiplexed, the saving for not allocating peak bandwidth increases. For Method I, bandwidth saving is only possible when $N \geq 10$. By considering the presence of low priority traffic in the mixed connections, Methods II-VI have pushed the limit down to less than 5 and these methods moreover require much smaller bandwidth as compared to Method I.

As shown in the figures, Method II requires the least bandwidth for any number of sources when $\eta = 0.15$, and for $N < 30$ when $\eta = 0.5$. However, the method may not necessarily satisfy the QoS required by the low priority traffic. In order to show this aspect, we find an optimised buffer threshold K_l, which is the largest threshold to just satisfy the QoS of the high priority traffic, using a bisection search, and plot the resulting loss probability for Methods II-VI obtained using the analysis in Section 3.4.4. The results are shown in Figure 3.16.



(a) $\eta = 0.15$

(b) $\eta = 0.5$

Figure 3.16 The corresponding cell loss probabilities versus the number of video sources for $QoS_l = 10^{-6}$ and $QoS_h = 10^{-9}$.

From the figures, we can see that only the QoS of high priority traffic is satisfied for Method II, while the low priority requirements are never satisfied for $\eta = 0.15$. The method overallocates the bandwidth for $\eta = 0.5$, which results in a much lower cell loss probability for low priority traffic than required when the number of sources is greater than 30. As the percentage of high priority traffic increases, we expect that the overallocation would become worse.

The Method VI provides the next least bandwidth requirement while meeting the QoS for both priority traffic. The use of priority multiplexing analysis in the method allows the loss probabilities for high and low priority traffic to be very close to the QoS requirements as can be seen from Figure 3.16. One drawback of this method is that it fails to satisfy the QoS for high priority traffic when $N < 5$. This is due to non-optimised buffer threshold values.

Methods III, IV, and V satisfy both QoS for every number of sources. Among them, Method V requires the least bandwidth. Methods III and IV, on the other hand, assign similar amount of bandwidths, which is due to the independence assumption of high and low priority cell arrivals. For correlated cell arrivals, we can expect that Method III will assign more bandwidth than Method IV because it can not take into account the statistical multiplexing among the traffic. The amount of bandwidth assigned by these methods strongly depends on the high priority traffic ratio η. The higher η value, the larger the assigned bandwidth. This results in a much lower cell loss probability for low priority traffic as shown in Figure 3.16. On the other hand, the bandwidth allocated by Method V seems to be less sensitive to the variation of η, yet it still satisfies the required QoS for both priority traffic.

Reducing the QoS for low priority cells reduces the bandwidth assigned by Methods III-VI, while Methods I and II are independence of this variation. Again Methods V and VI assigned the least bandwidth. Overall we can conclude that Method V is the best method as it assigns the least bandwidth while satisfying the QoS requirements for both high and low priority traffic for any number of sources.

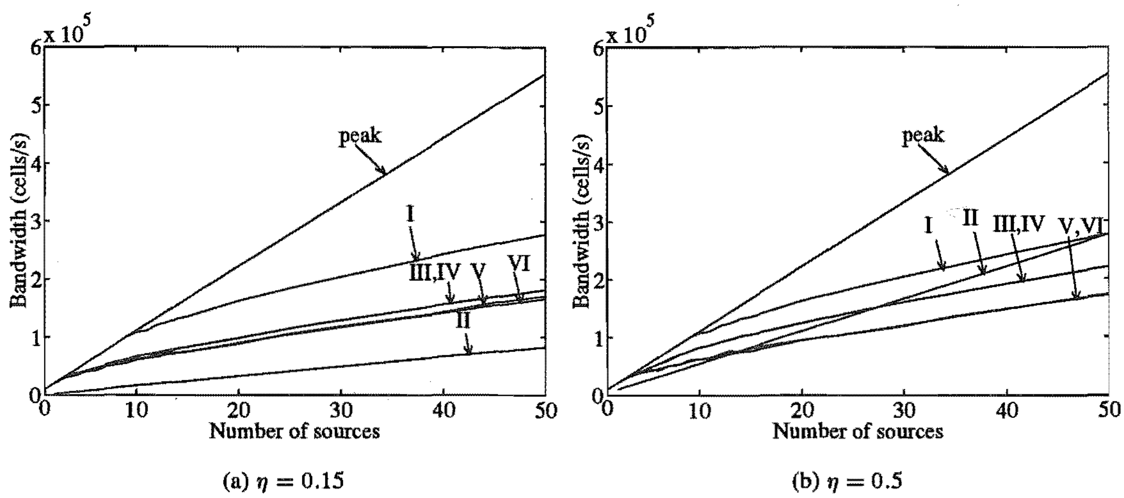


Figure 3.17 Assigned bandwidth versus the number of video sources for $QoS_l = 10^{-3}$ and $QoS_h = 10^{-9}$.

3.5.2 Heterogeneous Traffic Environment

In a heterogeneous traffic environment, the mix of N sources may have different characteristics. The mix can be represented by the tuple $(N_1, N_2, \ldots, N_I)$, where the N_i's are the number of sources of type i, and I is the number of distinct types of sources. Two approaches for determining the required bandwidth for a mix of N ($N = \sum_{i=1}^{I} N_i$) heterogeneous sources, generating high and low priority cells, can be identified as *class related rule (CRR)* and *aggregate traffic (AT)* approaches. The later approach is proposed here.

Class Related Rule Approach

The *class related rule (CRR)* approach was proposed by Gallassi *et al.* [1989]. According to this approach, bandwidth is assigned individually for each traffic class by Method I for pure connections, or by Method V for mixed connections. Let W_i be the bandwidth assigned for class i traffic under the assumptions that only class i traffic exists, based on the number of multiplexed connections and the traffic characteristic of the connections (peak rate, mean rate and mean burst length). Let W_T be the amount of bandwidth needed to guarantee the required QoS if the total traffic were offered by the "worst" class (i.e. the class with the highest burstiness), then the CRR approach assigns the bandwidth W by the following equation

$$W = \min(W_T, \sum_{i=1}^{I} W_i) \tag{3.66}$$

The CRR approach is applicable for any number of classes. The drawback of this approach is that the execution time of the procedure grows with the number of classes, since it requires as many bandwidth searches as the number of classes. Furthermore the approach only takes into account the statistical multiplexing gain within each traffic class, but not for the overall traffic.

Aggregate Traffic Approach

The *aggregate traffic (AT)* approach, proposed by us, takes full advantage of the statistical multiplexing among the traffic classes by using a procedure to match the statistical characteristic of the superposed sources to an MMPP source. It also requires a single search for the required bandwidth. The approach, following Method V, can be formulated as follows.

Algorithm 3.8.

Step 1. Calculate the mixture ratio η of high priority traffic to the total traffic using (3.2), i.e. $\eta = \sum_{i=1}^{I} p_i\lambda_i\eta_i / \sum_{i=1}^{I} p_i\lambda_i$.

Step 2. Determine the equivalent QoS requirement through the conservation law in (3.62), i.e. $QoS_e = \eta QoS_h + (1-\eta)QoS_l$.

Step 3. Assign initial values: $x_1 = \sum_{i=1}^{I} N_i p_i \lambda_i$ and $x_2 = \sum_{i=1}^{I} N_i\lambda_i$.

Step 4. Form $x_3 = (x_1 + x_2)/2$ and match the statistical characteristic of the superposition of heterogeneous sources to the MMPP parameters $(\lambda_1, \lambda_2, r_1, r_2)$ by the WS method.

Step 5. Evaluate the cell loss probability y_3 for a given K and with the above MMPP parameters by using analysis in Section 3.4.3.

Step 6. Substitute :
if $(y_3 > QoS_e)$ then do $x_1 \leftarrow x_3$; $y_1 \leftarrow y_3$ enddo.
else do $x_2 \leftarrow x_3$; $y_2 \leftarrow y_3$ enddo.

Step 7. Repeat steps 4-6 until $|(y_3 - QoS_e)/QoS_e| \le \epsilon$, where ϵ is the required precision (by default $\epsilon = 10^{-6}$).

Step 8. $W = x_3$.

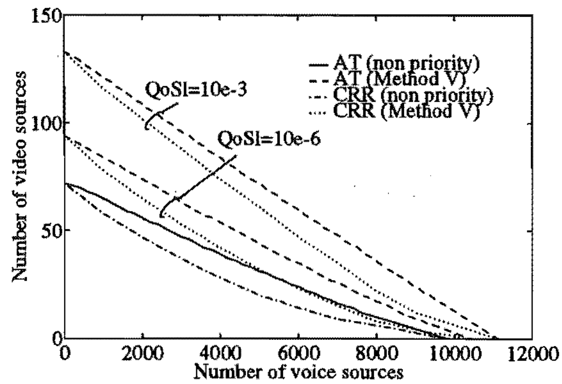
This approach is limited by the procedure for matching the statistical characteristics of the superposed sources to the MMPP parameters, which is not always applicable with a large number of classes. For example, the WS formulation is applicable to two classes of sources only. More general matching procedures, such as the LL method, are applicable to any number of classes. However, as shown in Section 3.3, this method gives less accurate results than the WS method.

Numerical Examples

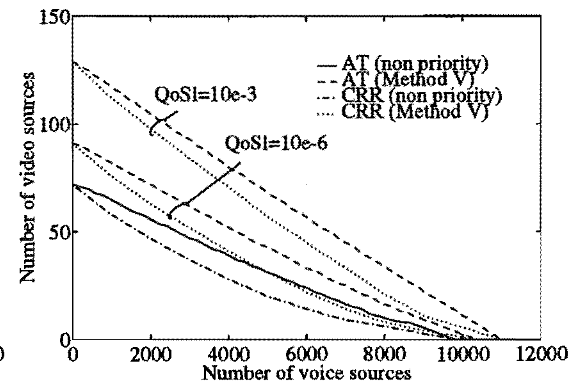
As an example, we consider the multiplexing of two classes of connections, namely voice and video, denoted here as class 1 and class 2, respectively. Each connection carries high and low priority cells with the fraction of high priority traffic to the total traffic being denoted by η. The QoS requirements are set at 10^{-9} for high priority traffic and 10^{-6} for low priority traffic. We intend to obtain an acceptance region which indicates the maximum number of connections of each class that can be accommodated by an ATM link with a capacity of 135.85 Mbps without violating the QoS requirements.

With the large number of voice connections that can be accomodated, the binomial distribution used in the WS method is not adequate, so we resort to the use of connection grouping, where we first group 10 voice connections to form a larger IPP source and then superpose the resulting sources. The parameter of the group source is related to the individual sources as

$$\lambda_{1g} = 10\lambda_1, \quad \alpha_{1g} = \alpha_1, \quad \beta_{1g} = \beta_1, \quad N_{1g} = N_1/10 \tag{3.67}$$



(a) $\eta = 0.15$



(b) $\eta = 0.5$

Figure 3.18 The acceptance regions for sources with the same η.

Figure 3.18(a) shows the acceptance regions for $\eta = 0.15$. The acceptance regions for the non-priority case are obtained by Method I. The figure shows that the introduction of cell loss priority in Method V for both CRR and AT approaches allows a larger number of sources to be accommodated compared with the non-priority case. As expected, the AT approach can accommodate larger number of sources than the CRR approach when two classes of sources are multiplexed, since it fully exploits the statistical multiplexing among the sources. The approaches admit the same number of sources when only sources of a single class are multiplexed. Comparing Figures 3.18(a) with 3.18(b), where $\eta = 0.5$, we can notice that the variation of η seems to have little effect on the results. The reason is that for a small value of QoS and for a difference between the high and low priority QoS greater than three orders of magnitude, $QoS_e \approx \eta QoS_l$, small QoS_l results in little difference between the resulting QoS_e for $\eta = 0.15$ and $\eta = 0.5$. This approximation also explains the large increase in the number of sources that can be accommodated as we vary the QoS requirements for low priority traffic from 10^{-6} to 10^{-3}. This increase implies also an increase in the utilisation of network resources as summarised in Table 3.3.

	Non-priority		Priority ($QoS_l = 10^{-6}$)		Priority ($QoS_l = 10^{-3}$)	
Utilisation	Peak	Method I	$\eta = 0.15$	$\eta = 0.5$	$\eta = 0.15$	$\eta = 0.5$
Minimum	0.165	0.372	0.485	0.470	0.687	0.666
Maximum	0.352	0.812	0.860	0.855	0.919	0.909

Table 3.3 Comparison of link utilisation for $\eta = 0.15$ and $\eta = 0.5$.

The minimum utilisation is observed when only video sources are multiplexed whereas the maximun utilisation is observed when only voice sources are multiplexed. The minimum and maximum utilisation for both CRR and AT approaches are the same since at those points only sources of a single class are multiplexed and the same underlying method, namely Method V, is used to find the required bandwidth in both approaches. From the table, we can observe that the use of cell loss priority plays a more significant role in increasing the utilisation of the video sources, which are more bursty, than the voice sources. As observed previously, the difference between the maximum and minimum utilisation for $\eta = 0.15$ and $\eta = 0.5$ is very small.

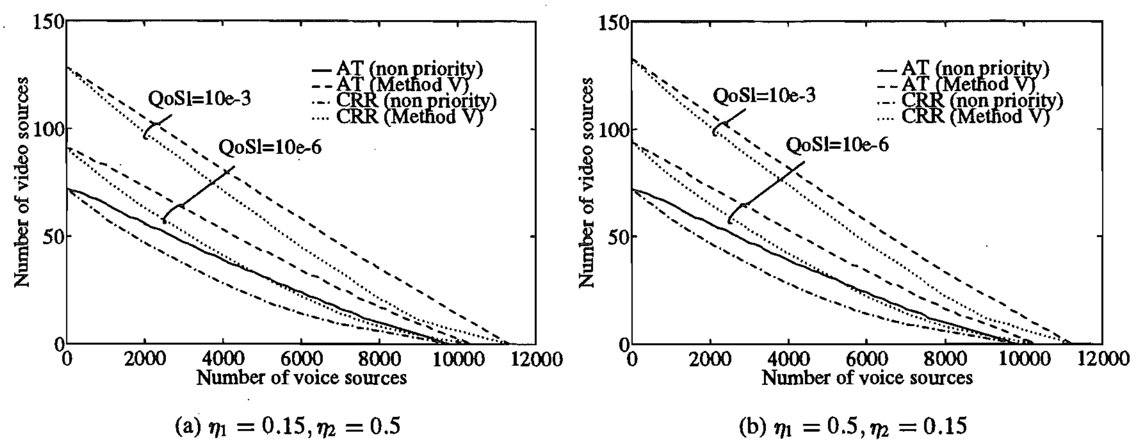


Figure 3.19 The acceptance regions for sources with different η.

Figure 3.19 plots the acceptance regions for the case where the fraction of high priority traffic in the voice sources is different from that in the video sources. The aggregate mixture ratio is calculated using (3.2). In general, the difference in η for both classes of sources results in an insignificant difference in the number of sources that can be accommodated. This again highlights the low sensitivity of the acceptance regions on the fraction of high priority traffic in each class.

3.6 Conclusion

In this Chapter, we have investigated the issue of bandwidth allocation for mixed connections. This involves modelling individual sources by an IPP model and approximating their superposition as an MMPP source. For matching the parameters from the superposed sources to the MMPP parameters, three methods were investigated and we found that the WS method [Wang and Silvester, 1993] performs best in both homogeneous and heterogenous traffic environments. The method is able to take into account the overload in the system and provides the closest approximation to the actual simulation of the IPP superposition.

Analysis of an ATM multiplexer fed by the resulting MMPP source, was carried out by assuming a discrete time system with a partial buffer sharing scheme. The analysis was verified by simulation. Two important relationships in the priority multiplexing system were also highlighted by the numerical results, namely, the conservation law and the invariance property of $\log_{10}(P_l/P_h)$ against varying buffer size as long as the fraction of high priority traffic η and the difference in the buffer threshold $K - K_l$ remain constant.

Based on these two relationships, we developed two methods, namely Methods V and VI, for allocating bandwidth to mixed connections in a homogeneous traffic environment and compared their performance with that of existing methods. Method VI was shown to assign the least bandwidth as it allows the loss probabilities for high and low priority traffic to be very close to the QoS requirements. It can also fully exploit the statistical multiplexing between the high and low priority traffic. The method, however, does not always satisfy the QoS requirements for high priority traffic, especially when only a small number of sources are multiplexed. Overall, Method V was shown to be the best for assigning the least bandwidth, while satisfying the QoS for both high and low priority traffic with any number of sources. The method depends weakly on the fraction of high priority traffic to the total of traffic, and strongly on the variation of the QoS requirements for low priority traffic.

For allocating bandwidth to mixed connections in a heterogeneous traffic environment, we have proposed the *aggregate traffic (AT)* approach which exploits statistical multiplexing among the connections of different classes in addition to the statistical multiplexing among the connections of the same class as in the *class related rule (CRR)* approach. The approach allows a larger number of voice and video connections to accommodated than the CRR approach when connections of both classes are multiplexed. Again, it was found that the boundaries of the acceptance regions are less sensitive to the variation of the the fraction of high priority traffic to the total traffic. They only depend on the QoS requirement of the low priority traffic.

In summary, the reduction in the bandwidth required due to the introduction of cell loss priority has been highlighted in this Chapter. This reduction is beneficial for the users to reduce their communication costs and for the network to increase the number of sources that can be supported, as demonstrated by the acceptance regions for heterogeneous traffic conditions. In order to reflect this reduction of bandwidth, the mechanims used for policing mixed connections should be able to take into account the presence of pretagged cells in the connections. This issue of policing mixed connections are discussed in the next Chapter.

Chapter 4

POLICING MIXED CONNECTIONS

In the previous Chapter, we have investigated various methods for the purpose of properly characterising the effective bandwidth of an individual mixed connection for connection admission control. After a connection has been admitted, there is a need to ensure that the traffic generated by the source remains consistent with the negotiated traffic parameters, in order to ensure that the network operates free of congestion. For this reason, a policing function, (also referred to as a usage parameter control (UPC)), has been suggested for regulating the cell input flow at each access node. A large number of policing schemes have been proposed, as summarised in Section 1.5.3. The leaky bucket algorithm is the basis for the most popular schemes and has been widely studied, using simulation as well as analytical techniques, for various input traffic patterns. A review of these studies is given in Section 4.1.

Most existing leaky bucket schemes assume that all cells submitted by users have the same significance, so that no preference is given in discarding, buffering or marking the cells if the negotiated traffic parameters are exceeded. Such a uniform treatment may not be appropriate for policing mixed connections since the presence of cells pretagged as low priority by users is not taken into account. Therefore our objective in this Chapter is to investigate modifications of existing leaky bucket schemes for more effectively policing mixed connections. We derive expressions for cell discard rate or marking rate, using discrete-time based analysis for both existing and new schemes and compare their performance in policing individual sources generating bursty traffic with pretagged cells (e.g. video sources). In addition to studying the leaky bucket schemes in isolation, we also study the performance of a multiplexer fed by a number of policed sources and the quality of service experienced by a source via simulation.

In order to make sense of the large number of existing leaky bucket schemes and to uncover new schemes, we begin by proposing two classifications in Section 4.2. In Section 4.3, we present a discrete-time based analysis of the IPP/D/1/K queue, which will be extensively applied for analysing the performance of various leaky bucket schemes used for policing an IPP source generating pretagged cells. We describe various existing leaky bucket schemes in Section 4.4, followed in Section 4.5 by the description and analysis of six modified leaky bucket schemes proposed for teletraffic with pretagged cells in mixed connections. Verification of the analytical results are carried out in Section 4.6 by a comparison with simulation results. In Section 4.7, we compare the cell discard rate of the leaky bucket schemes and the effect the schemes have on the end-to-end cell loss probability of a connection. Section 4.8 concludes this Chapter with a summary of performance comparison results.

4.1 Review of Related Work

The behaviour of an isolated leaky bucket has been investigated in many studies under different assumptions regarding models of cell and token arrivals. For example, Hughes *et al.* [1990a] assumed that cell arrivals are governed by a Bernoulli process, Rathgeb [1991] and Butto *et al.* [1991] evaluated cell loss probabilities assuming a two-state on/off arrival process, Sohraby and Sidi [1991] considered the m-state MMPP input process, Berger [1991] analysed the leaky bucket performance assuming a Markovian arrival process (MAP) for cell arrivals and a renewal process for token generation, while Rathgeb [1992] studied the use of the original leaky bucket for policing the recorded video information.

A general consensus indicates that the original leaky bucket, which simply discards excess cells, requires large safety margins (in terms of the required bucket depth and token rate) in order to effectively police the mean rate of a connection. This is especially true for bursty traffic, such as still picture or video traffic, where esentially only peak policing is possible [Rathgeb, 1991]. For this reason, improvements to the leaky bucket scheme were suggested by introducing buffering [Sidi *et al.*, 1989], marking the excess cells [Eckberg *et al.*, 1989; Gallassi *et al.*, 1990a] or a combination of both [Bala *et al.*, 1990].

Performance comparisons between the original leaky bucket and ones with buffering or marking have also been carried out. However, less bursty traffic, such as voice traffic, has been commonly assumed. Gravey *et al.* [1991] compared the leaky bucket with marking scheme with the original leaky bucket scheme by assuming that the marked traffic and the unmarked traffic are two independent Poisson processes. Chao [1991] did a performance comparison of the original leaky bucket and the buffered leaky bucket with and without marking for an IPP source model and concluded that the leaky bucket in which violated cells are buffered or marked offers better performance than the original leaky bucket, and the leaky bucket which combines both buffering and marking policies performs the best in terms of the cell discard rate.

With the possibility of users being allowed to pretag cells as being low priority within a single connection, the need for a leaky bucket that takes into account this pretagged traffic led Waterman, Hartanto and Sirisena [1993] to propose the leaky bucket for pretagged traffic scheme. Its performance was compared with the original and buffered leaky bucket schemes, assuming a superposition of voice sources. The sources were modelled according to a voice coding technique proposed by Sriram *et al.* [1991], which separated the most significant and the least significant bits from a voice sample into two cells, tagged as high and low priority, respectively. Simulation results showed that making provision for pretagged traffic in the policing mechanism can yield three orders of magnitude improvement in the loss probability of high priority cells as compared to the original leaky bucket. However, this improvement may be overstated since the multiplexing of voice sources prior to the policing action results in a smoother input traffic, which means that the effect of bursty traffic was not fully accounted for in [Waterman *et al.*, 1993].

The effects of statistical multiplexing policed sources within an ATM network were studied both analytically and by simulation. Simulation studies were reported in [Friesen and Wong, 1993; Hemmer and Huth, 1991; Monteiro *et al.*, 1991]. On the other hand, analytical results were obtained through characterising the output process from the leaky bucket and then treating this output process as the input process to the multiplexer. Ren *et al.* [1994] provided an exact analysis of the queueing model using a discrete time analysis, while Elwalid and Mitra [1991] analysed the output process from a leaky bucket with marking scheme and approximated their superposition by a source model based on a Markov modulated fluid flow (MMFF) as an input to an ATM multiplexer. The multiplexing of policed bursty sources that carry pretagged traffic does not appear to have been investigated previously and therefore it forms an important topic in the present study.

4.2 Classification of Leaky Bucket Schemes

In the following discussion, we use the terms *unmarked* and *marked* to refer to the high and low priority cells created by a marking scheme, while reserving the terms *untagged* and *tagged* (or *pretagged*) for referring to the high and low priority cells generated by a source.

4.2.1 Single Bucket and Dual Bucket Classes

By having pretagged and marked traffic entering the network, the leaky bucket schemes can (i) police the traffic separately from untagged and unmarked traffic or (ii) police the pretagged traffic along with the untagged traffic or (iii) leave the traffic unpoliced. An additional bucket is required in the former case. Therefore, based on the presence or absence of this additional bucket, we can differentiate leaky bucket schemes into two classes, namely

Single bucket. A leaky bucket of this class utilises a single token pool. For a leaky bucket without marking scheme, the transmission capability of sources is limited up to the token generation rate. On the other hand, for a leaky bucket with marking scheme, the transmission capability for low priority traffic is unlimited, which can cause the traffic to overload the network and to degrade the performance of high priority traffic if they partially share some network resources [Hartanto *et al.*, 1991].

Dual bucket. A leaky bucket of this class utilises two token pools to separately police the high and low priority (pretagged and marked) traffic rates. For a leaky bucket without marking scheme, this additional bucket allows users to send more traffic than negotiated as pretagged traffic in order to make use of any idle bandwidth within the network and at the same time limits the amount of pretagged traffic that can enter the network. For a leaky bucket with marking scheme, this additional bucket limits the number of marked cells that are allowed to enter the network.

This class does not include a cascaded leaky bucket (also referred to as a dual leaky bucket) [Yamanaka *et al.*, 1993], where two leaky buckets are placed in tandem to police both the peak rate and the average rate of a connection, as following our classification such scheme is a combination of two leaky buckets from the single bucket class.

4.2.2 Non-Priority and Priority Classes

Depending on the treatment of pretagged cells in a mixed connection, we can also differentiate leaky bucket schemes into two classes, namely

Non-Priority. Leaky buckets of this class treat a mixture of low and high prioirty cells in a uniform way. This implies that untagged and pretagged cells of a given connection have the same chance of being discarded or marked if no token is available.

Priority. Leaky buckets of this class take into account the presence of pretagged cells in a stream of cells and provide better treatment for untagged cells than for pretagged cells, for example, by reserving larger input buffer space or more tokens for them, hence reducing the cell discard rates of the untagged cells.

Following this classification, Table 4.1 lists all leaky bucket schemes which are considered in this Chapter, pointing additionally Sections in which the schemes are described in detail. The acronyms used follow the format $x_1x_2LBy_1y_2$, where LB(leaky bucket) forms the base, with the prefixes x_1=B(buffered); x_2=D(dual); and the suffixes y_1=M(marking); y_2=P(priority). For example, BDLBMP stands for a buffered dual leaky bucket with marking and priority. Not all schemes have all prefixes or suffixes. For example, the leaky bucket with marking (LBM) only has one suffix. A special acronym OLB is used to indicate the original leaky bucket proposed by Turner [1986], while BDLBP-DT is used for buffered dual leaky bucket with priority and dynamic threshold. BDLBP-DT has been proposed in this Chapter and it follows a new concept in policing mixed connections rather than a simple modification of existing leaky bucket schemes.

	Single or	Policing mechanisms		
	Dual (S/D)	Discarding	Buffering	Marking
Non-Priority class:				
OLB (Section 4.4.1)	S	✓		
BLB (Section 4.4.2)	S	✓	✓	
LBM (Section 4.4.3)	S			✓
DLBM (Section 4.4.4)	D	✓		✓
BLBM (Section 4.4.5)	S		✓	✓
BDLBM (Section 4.4.6)	D	✓	✓	✓
Priority class:				
DLBP (Section 4.5.1)†	D	✓		
BLBP (Section 4.5.2)	S	✓	✓	
BDLBP (Section 4.5.3)†	D	✓	✓	
BDLBP-DT (Section 4.5.4)†	D	✓	✓	
DLBMP (Section 4.5.5)	D	✓		✓
BDLBMP (Section 4.5.6)†	D	✓	✓	✓

†newly proposed schemes.

Table 4.1 Leaky bucket schemes.

In the table, we also listed three possible policing mechanisms for the leaky bucket schemes. For a single bucket class, the marking mechanism is an alternative to the discarding mechanism. On the other hand, for a dual bucket class, all mechanisms are applicable for the first bucket, but only discarding mechanism is applicable for the second bucket. For this reason, not all combinations of the single bucket/dual bucket classification and the three policing mechanisms are possible.

On the other hand, following the non-priority/priority classification, in theory, there are seven basic leaky bucket schemes in non-priority case and seven in priority case based on all possible combinations of the three policing mechanisms. This excludes the case with no policing mechanisms in either class. So far, we have only listed six schemes for non-priority class and six schemes out of four combinations for priority class (referring to the table, one can notice that BLBP, BDLBP, and BDLBP-DT employ the same combination). The remaining combination for non-priority class is the leaky bucket with buffering, which is also known as *traffic shaper*. In the scheme, cells are allowed to queue indefinitely in a buffer when no token is available, hence no discarding or marking of cells occurs. The scheme is normally used at the user ends rather than at the access nodes as for other leaky bucket schemes listed in Table 4.1 and for this reason, it will not be considered in this Chapter.

The remaining combinations for priority class are the leaky bucket with buffering only, the leaky bucket with marking and priority (LBMP), and the buffered leaky bucket with marking and priority (BLBMP). The leaky bucket with buffering only can be modelled as BDLBP with infinite buffer. This scheme will not be considered as the location of the policer is different from other leaky buckets investigated in this Chapter. On the other hand, the last two schemes (LBMP and BLBMP) are special cases of DLBMP and BDLBMP, respectively, where the marked and pretagged cells are not discarded. These schemes will not be considered in this Chapter, as marking by the network is not of primary interest in this thesis. Moreover the performance of these schemes have been represented by DBLMP and BDLBMP. Consequently it is unnecessary to consider the performance of these special cases separately.

4.3 Mathematical Preliminaries

In studying various leaky bucket schemes, we will assume that the source is described by an IPP model and we are primarily concerned with the cell discard rate of a leaky bucket. We assume that cell delay requirements can be satisfied by choosing the right buffer size. The following performance measures and variables for leaky bucket schemes are used.

λ, α, β	the IPP parameters defined in Section 3.2.1.
η	the ratio of high priority traffic to total traffic in a mixed connection.
γ	the token arrival rate in a leaky bucket of the single bucket class.
γ_1, γ_2	the arrival rates of tokens to the first and the second token pool in a leaky bucket of the dual bucket class, respectively.
B	the bucket depth in a leaky bucket of the single bucket class.
B_1, B_2	the bucket depths of the first and the second token pool in a leaky bucket of the dual bucket class, respectively.

f_R	the normalised token rate, namely $f_R \triangleq \gamma / R_m$, where R_m is the mean traffic rate of a source.
f_{R1}, f_{R2}	the normalised token rates for the first and the second token pool in a leaky bucket of the dual bucket class, respectively.
f_S	the split factor or the ratio between the token rate and bucket depth of the first token pool to the total token rate and bucket depth in a leaky bucket with priority of the dual bucket class, namely $f_S \triangleq f_{R1}/f_R$, where $f_R = f_{R1} + f_{R2}$ and also $f_S \triangleq B_1/B$, where $B = B_1 + B_2$.
K', K'_l	the input buffer thresholds of a leaky bucket.
P_{d1}, P_{d2}	the cell discard rate for the first and the second token pool of a leaky bucket of the dual bucket class.
$P_{d(x)}$	the cell discard rate of the x scheme, e.g. $x = OLB$.
$P_{dh(x)}, P_{dl(x)}$	the loss probability experienced by high and low priority cells, which are policed by the x scheme.
$P_{m(x)}$	the cell marking probability of the x scheme.

In the list the coefficient f_S is called the *split factor*, because it reflects the splitting of tokens between the first and the second bucket in a priority case. It is introduced for the purpose of comparing the performance of non-priorty case to priority case regardless the absolute value of token rates in both cases. Similarly, the coefficient f_R is introduced for the purpose of comparing the performance among all schemes regardless the absolute value of mean traffic rate of a source and the token rate.

The analysis of an IPP/D/1/K queue, which will be used extensively for deriving the performance measures, can be derived from the MMPP/D/1/K analysis given in Section 3.4, with the OFF and ON states of the IPP sources corresponding to the State 1 and State 2 of the MMPP model, respectively. Let $q_1^-(n,k,j)$ and $q_1^+(n,k,j)$ denote the probabilities that during the nth cycle, there are j cells in the queue just prior to, and just after the beginning of the kth slot for State 1, respectively. Following the MMPP/D/1/K analysis with $\lambda_1 = 0$, $\lambda_2 = \lambda$, $r_1 = \alpha$, $r_2 = \beta$, being normalised to the output link rate, we have the probability mass functions for the queue length immediately prior to and immediately after the beginning of a time slot for State 1 and State 2 as

$$\begin{aligned} q_1^+(n,k,j) &= q_1^-(n,k,j) && \text{if } 0 \le j \le K+1 \\ q_1^-(n,k+1,j) &= \Sigma_0(q_1^+(n,k,j+1)) && \text{if } 0 \le j \le K \\ q_2^+(n,k,j) &= \Sigma^{K+1}(q_2^-(n,k,j) * a_2(j)) && \text{if } 0 \le j \le K+1 \\ q_2^-(n,k+1,j) &= \Sigma_0(q_2^+(n,k,j+1)) && \text{if } 0 \le j \le K \end{aligned} \tag{4.1}$$

where

$$a_2(j) = \lambda_2^j \exp(-\lambda_2)/j\,! \tag{4.2}$$

The relationship between State 1 and State 2 follows (3.39) and (3.40) with $s_1(k) = r_1(1 - r_1)^{k-1}$ and $s_2(k) = r_2(1 - r_2)^{k-1}$. Under the steady state condition, we have

$$q_1^-(k,j) = \lim_{n\to\infty} q_1^-(n,k,j) \tag{4.3}$$

In deriving the cell loss probability, we follow (3.49) in Section 3.4.3. Let denote L_1 and L_2 as the average cells lost when the source is in State 1 and State 2, respectively. Since there is no arrival during State 1, $L_1 = 0$ and the equation is reduced to

$$P_{IPP/D/1/K} = \frac{L_2}{\lambda_2} = \sum_{k=1}^{\infty} s_2(k) \sum_{j=0}^{K} q_2^-(k,j) \sum_{i=K-j+2}^{\infty} [i-(K-j+1)] \frac{a_2(i)}{\lambda_2} \quad (4.4)$$

4.4 Non-Priority Class of Leaky Buckets

The leaky bucket of this class does not differentiate the presence of pretagged cells in mixed connections, therefore the loss probabilities or the marking probabilities are the same for both the untagged and the pretagged cells.

4.4.1 Original Leaky Bucket (OLB)

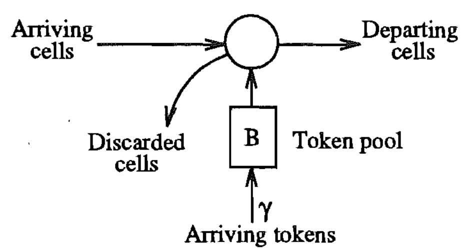


Figure 4.1 The original leaky bucket (OLB).

The *original leaky bucket (OLB)*, depicted in Figure 4.1, was proposed by Turner [1986]. In the scheme, tokens are generated at a fixed rate γ tokens per slot, or one token every $D = 1/\gamma$ slots, and stored in a token pool that can hold a maximum of B tokens. An arriving cell requires a token to enter the network. If no token is available, then the cell will be discarded. The parameters γ and B fully describe the scheme.

Basic Properties

We assume that the time axis is slotted, each time slot being equal to the time required for transmitting one cell, and that a token is generated every D slots. If initially there are X tokens in the pool, then the maximum number of tokens at the end of an interval of ND slots is equal to $X+N$ and the maximum number of cells that can be transmitted during that interval is equal to

$$\min(X+N, ND) \quad (4.5)$$

This results in an instantaneous rate of cells entering the network as being equal $\min(X+N, ND)/(ND)$ cells per slots. The long term average of this rate can be obtained by taking the limit as $N \to \infty$

$$\lim_{N\to\infty} \frac{\min(X+N, ND)}{ND} = \frac{1}{D} = \gamma, \quad (4.6)$$

which indicates that the long term mean rate of cells entering the network is bounded by the token generation rate.

On the other hand, the maximum number of cells, that enter the network consecutively without any loss, is determined by the token pool. This takes place if the burst arrives at the time when the token pool is full. During the transmission of B cells of the burst, B/D new tokens arrive. These new tokens allow additional transmissions of B/D cells and during these transmissions, a further $(B/D)/D$ new tokens arrive, and so on. So, the maximum length of burst allowable is equal to

$$B + \frac{B}{D} + \frac{B}{D^2} + \frac{B}{D^3} + \ldots = \frac{B}{(1 - 1/D)} = \frac{B}{(1 - \gamma)} \tag{4.7}$$

Mathematical Analysis

It is generally known that the exact analysis of the ordinary leaky bucket is equivalent to the analysis of G/D/1/K queueing system through the concept of duality with the condition of empty token pool being equivalent to the condition of the full buffer in G/D/1/K [Guerin *et al.*, 1991]. Thus, the performance of the leaky bucket for policing an IPP source can be analysed using IPP/D/1/K queue as the model. For an OLB, the IPP parameters $\lambda_2 = \lambda$, $r_1 = \alpha$, and $r_2 = \beta$ should be normalised to the token rate γ of the leaky bucket and with $K = B$, we can obtain the cell discard rate as

$$P_{d(OLB)} = P_{IPP/D/1/B} \tag{4.8}$$

Drawbacks of OLB Scheme

Ideally, a policing scheme should always take action when the negotiated traffic parameters are violated, but remaining completely transparent to compliant traffic. In practice, two types of enforcement errors will arise [Le Boudec, 1992; Hughes *et al.*, 1992]

Error I: Cells may be discarded even though the negotiated traffic parameters are not violated. This occurs because of the statistical fluctuation of the traffic and the variable transfer delays experienced by cells between the source and the policing unit.

Error II: Cells may be accepted although the negotiated traffic parameters are violated. This occurs because of some safety margins allowed by the policing scheme in order to minimise Error I.

Recent studies of the OLB [Butto *et al.*, 1991; Rathgeb, 1991] show that protection against Error I requires such large safety margins for the leaky bucket parameters that this is not effective in protecting against Error II. In order to reduce the required safety margins, more sophisticated policing algorithms have been proposed. They basically defer the discarding of violating cells by either marking or buffering the cells or a combination of both as will be discussed in the following sections.

4.4.2 Buffered Leaky Bucket (BLB)

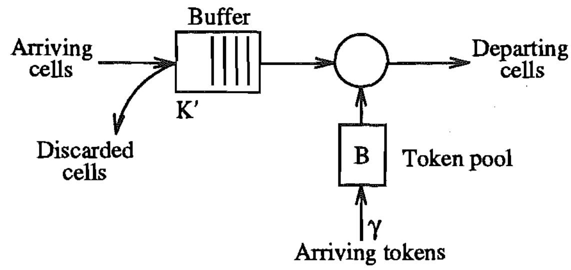


Figure 4.2 The buffered leaky bucket (BLB).

Figure 4.2 shows the *buffered leaky bucket (BLB)* [Woodruff *et al.*, 1988; Sidi *et al.*, 1989; Berger, 1991] which differs from OLB in the addition of a buffer of size K'. Instead of discarding arriving cells that find no tokens available, BLB buffers the cells. If only the buffer is full, then the arriving cells are discarded. The buffer length can be calculated to meet the maximum delay requirements for the connection as the buffer simply imposes a time delay, $K'D$. The scheme degenerates to OLB if $K' = 0$.

The BLB can be analysed as a G/D/1/K queue [Sidi *et al.*, 1989]. Thus, the cell discard rate of a BLB for policing an IPP source can be obtained from the IPP/D/1/K analysis with $K = K' + B$ and it is given as

$$P_{d(BLB)} = P_{IPP/D/1/K'+B} \tag{4.9}$$

The equation indicates that the cell loss probability only depends on $K' + B$ and not individually on K' or B. The mean cell delay at the leaky bucket, however, decreases as B increases, if $K' + B$ is kept constant [Chuah and Cruz, 1990].

4.4.3 Leaky Bucket with Marking (LBM)

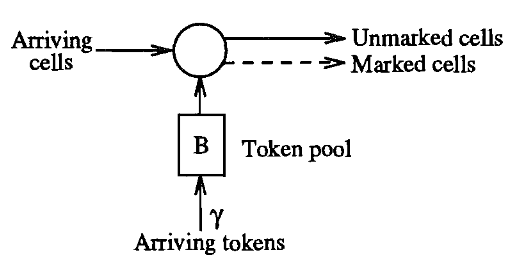


Figure 4.3 The leaky bucket with marking (LBM).

The *leaky bucket with marking (LBM)*, proposed by Eckberg *et al.* [1989] and Gallassi *et al.* [1989], is depicted in Figure 4.3. In this scheme an arriving cell passes through the leaky bucket as an unmarked cell if it obtains a token from the token pool, otherwise the cell is marked as low priority. Since no cells are dropped in this case, the cell loss probability at the leaky bucket is zero. The number of cells being marked in LBM is the same as the number of cells lost in OLB for the same parameters of leaky bucket and source. Thus if $P_{m(LBM)}$ means the probability of a cell being marked under LBM, then

$$P_{m(LBM)} = P_{d(OLB)} = P_{IPP/D/1/B} \tag{4.10}$$

The marked cells that enter the network will be discarded if congestion arises. With current standardisation of a single bit for cell loss priority (CLP) [CCITT, 1992a], no distinction is possible between the marked cells under LBM and the pretagged cells in mixed connections within the network. High priority cells (from the source's point of view) that are marked at the leaky bucket have the same chance of being discarded as low priority cells. Since any loss of high priority cells containing the most significant bits in speech or video coding can cause sudden degradation of service quality, it is important to estimate how many high priority cells are marked and lost within the network and this will be our focus in simulation studies when we compare the performance of LBM with other leaky bucket schemes.

4.4.4 Dual Leaky Bucket with Marking (DLBM)

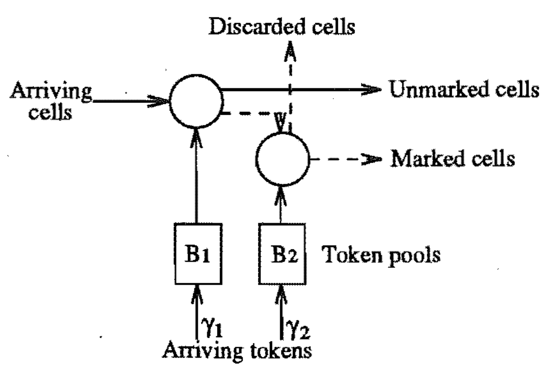


Figure 4.4 The dual leaky bucket with marking (DLBM).

In LBM, the marked cells that enter the network are uncontrollable. If the limitation on the marked cells is necessary, then additional token pool can be used, resulting in a *dual leaky bucket with marking (DLBM)* shown in Figure 4.4. In this scheme an arriving cell passes through the leaky bucket as an unmarked cell if it obtains a token from the first token pool, otherwise the cell is marked as low priority. The marked cell will enter the network if it obtains a token from the second token pool, otherwise it will be discarded.

In analysing the scheme, we firstly notice that the first bucket acts as an LBM. Hence utilising the analysis for LBM, which in turns follows the analysis of OLB, with $K = B_1$ and the IPP parameters being normalised to the token rate γ_1 of the first bucket, we can estimate the cell marking probability for DLBM as

$$P_{m(DLBM)} = P_{IPP/D/1/B_1} \tag{4.11}$$

The marked cells from the first bucket are policed by the second bucket. We assume that the arrivals of the marked cells to the second bucket are approximated by the original IPP sources with $\lambda_2 = \lambda P_{m(DLBM)}$, $r_1 = \alpha$, and $r_2 = \beta$ being normalised to the token rate γ_2 of the second bucket. The cell discard rate P_{d2} at the second bucket can then be determined from the IPP/D/1/K analysis with $K = B_2$, i.e.

$$P_{d2} = P_{IPP/D/1/B_2} \tag{4.12}$$

Since the cells have to be marked first before they are policed by the second bucket and discarded, we must condition P_{d2} by the marking probability $P_{m(DLBM)}$ in order to find an overall cell discard rate of a DLBM, i.e.

$$P_{d(DLBM)} = P_{d2}\, P_{m(DLBM)} \tag{4.13}$$

The accuracy of the approximation will be assessed by using simulation in Section 4.6.

4.4.5 Buffered Leaky Bucket with Marking (BLBM)

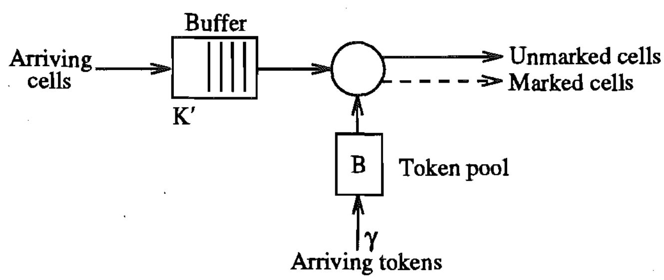


Figure 4.5 The buffered leaky bucket with marking (BLBM).

In LBM, an arriving cell is marked if it finds no available token. As delivery of the marked cell is dependent on the current network loading and can not be guaranteed, it is preferable to defer the marking of the cell as long as possible through buffering the cell. The *buffered leaky bucket with marking (BLBM)* [Chao, 1991] facilitates this approach through the additional buffer as shown in Figure 4.5.

In the scheme, an arriving cell will enter the network if there is a token in the pool and no cells are waiting in the buffer, otherwise it will be queued. If the buffer is full, then the cell at the head of the buffer is chosen and marked as low priority and sent out to make way for the incoming cell. We cannot mark the newly arriving cell as this will lead to the cell being delivered out of sequence, if there are some earlier arrived cells in the buffer. This indicates some complexity in handling the buffering process.

Following the BLBM, the number of marked cells entering the network is unlimited, hence the cell loss probability for this scheme is zero. The cell marking probability of the scheme is the same as the cell loss probability in BLB, thus

$$P_{m(BLBM)} = P_{d(BLB)} = P_{IPP/D/1/K'+B} \tag{4.14}$$

4.4.6 Buffered Dual Leaky Bucket with Marking (BDLBM)

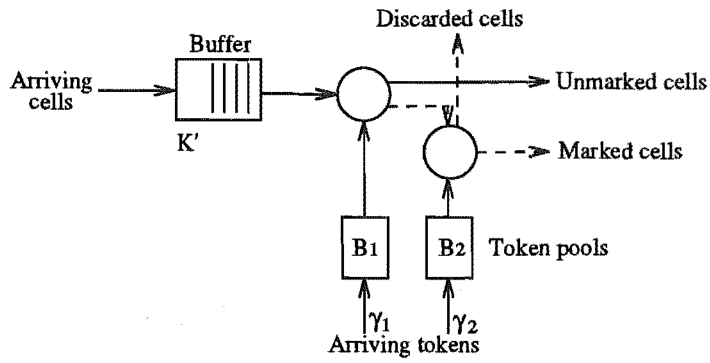


Figure 4.6 The buffered dual leaky bucket with marking (BDLBM).

The *buffered dual leaky bucket with marking (BDLBM)*, depicted in Figure 4.6, can be viewed as either an improvement of BLBM by limiting the marked traffic rate admitted to the network or a modification of DLBM based on introduction of the additional input buffer. In the scheme, an arriving cell is buffered if there is no token in the first token pool or the buffer is not empty. If the buffer is full, then the cell at the head of the buffer is chosen and marked as low priority to make way for the incoming cell. The marked cell will enter the network if it obtains a token from the second token pool, otherwise it will be discarded.

A similar scheme, called a *generalised leaky bucket*, was proposed by Bala *et al.* [1990] for policing data sources with variable message length in the PARIS network, instead of fixed cell length as for the ATM network. The scheme, however, has not been analysed previously.

Due to the buffering process, an exact analysis of BDLBM is not possible without actually analysing the output process of the first bucket, hence an approximate solution will be given. The solution follows a similar procedure as for the DLBM solution. We first find the cell marking rate for the first bucket, which is virtually the same as the cell discard rate of a BLB with the IPP parameters being normalised to the token rate γ_1 of the first bucket, i.e.

$$P_{m(BDLBM)} = P_{d(BLB)} = P_{IPP/D/1/K'+B_1} \tag{4.15}$$

By assuming that the arrivals of the marked cells to the second bucket are approximated by the stream generated by the original IPP source with $\lambda_2 = \lambda P_{m(BDLBM)}$, $r_1 = \alpha$, and $r_2 = \beta$ being normalised to the token rate γ_2 of the second bucket, we can find the cell discard rate P_{d2} at the second bucket as

$$P_{d2} = P_{IPP/D/1/B_2} \tag{4.16}$$

Since no cells will be discarded unless the input buffer is full, we must condition P_{d2} upon the probability of the buffer being full, which is the same as $P_{m(BDLBM)}$. The overall cell discard rate is equal to

$$P_{d(BDLBM)} = P_{d2}\, P_{m(BDLBM)} \tag{4.17}$$

The accuracy of the approximation will be assessed using simulation in Section 4.6.

4.5 Priority Class of Leaky Buckets

Leaky buckets of this class have been proposed for differentiating untagged and pretagged cells transmitted in mixed connections. The performance of such a scheme is measured by the cell discard rate for high priority (untagged and unmarked) traffic P_{dh} and low priority (pretagged and marked) traffic P_{dl}. To our knowledge, such schemes (except the BLBP and the DLBMP) have not been considered previously.

4.5.1 Dual Leaky Bucket with Priority (DLBP)

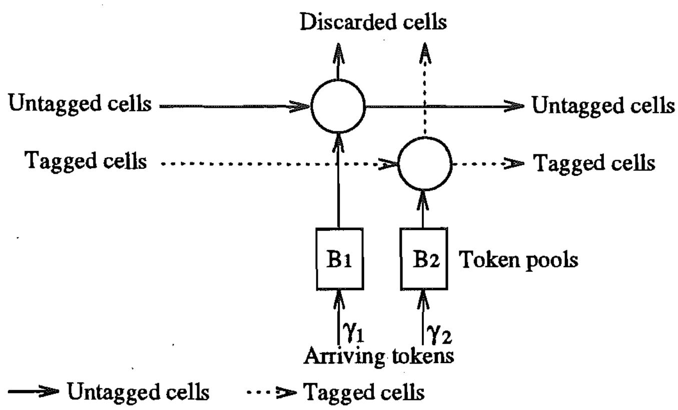


Figure 4.7 The dual leaky bucket with priority (DLBP).

The *dual leaky bucket with priority (DLBP)*, depicted in Figure 4.7, is a modification of OLB for the situation where both untagged and tagged traffic is present. It comprises two token pools for separate policing the rate of untagged and tagged traffic. The first token pool is for policing untagged traffic and the second one is for policing tagged traffic. Each token pool has its own parameters, indicated here as B_1 and γ_1 for bucket depth and token rate of the first bucket, and B_2 and γ_2 for the second one. The additional bucket provides more degree of freedom for separately controlling the traffic rates, thus we can enhance the services offerred to untagged traffic by selecting larger values for the parameters B_1 and γ_1 than B_2 and γ_2, given the same total token rate ($\gamma = \gamma_1 + \gamma_2$) and total bucket depth ($B = B_1 + B_2$) as in OLB. In analysing the scheme, we assume that cells of the untagged and tagged traffic arrive independently to their respective buckets and therefore DLBP can be viewed as a combination of two separate OLBs. The first bucket is assumed to police an IPP source with $\lambda_2 = \eta\lambda$, $r_1 = \alpha$, and $r_2 = \beta$, where η is the ratio of high priority traffic intensity to the overall traffic intensity. Utilising IPP/D/1/K analysis with the IPP parameters being normalised to the token rate γ_1 and $K = B_1$, we can find the cell discard rate for untagged traffic as

$$P_{dh(DLBP)} = P_{IPP/D/1/B_1} \tag{4.18}$$

The analysis of the second bucket follows the same procedure as above with $\lambda_2 = (1 - \eta)\lambda$, $r_1 = \alpha$, $r_2 = \beta$ being normalised to the token rate γ_2 of the second bucket and $K = B_2$. Thus, the cell discard rate for pretagged traffic is

$$P_{dl(DLBP)} = P_{IPP/D/1/B_2} \tag{4.19}$$

4.5.2 Buffered Leaky Bucket with Priority (BLBP)

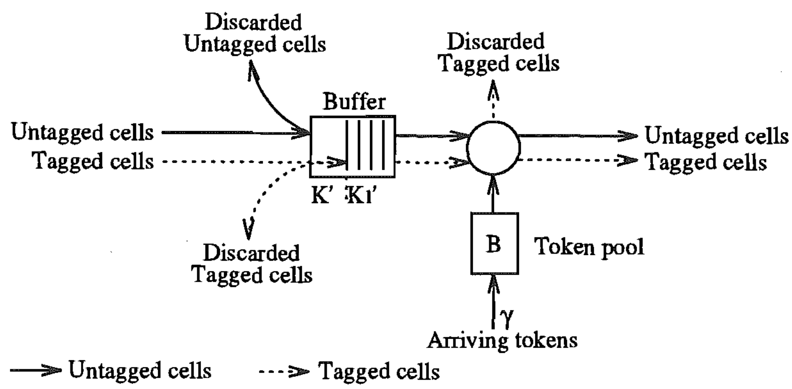


Figure 4.8 The buffered leaky bucket with priority (BLBP).

The *buffered leaky bucket with priority (BLBP)*, depicted in Figure 4.8, was proposed by Waterman, Hartanto and Sirisena [1993]. This scheme can be considered as a modification of a BLB for handling pretagged traffic, introduced for minimising the loss of untagged cells at the expense of the pretagged cells. In order to achieve this aim, an additional threshold K_l' in the input buffer is introduced to limit access to the buffer by pretagged cells.

The operation of the scheme is as follows. An arriving pretagged cell will be buffered if the queue is less than K_l', otherwise it will be discarded. When the pretagged cell reaches the head of the queue, it will obtain a token and enter the network if the token pool is not empty, otherwise it will be discarded in order to prevent the cell from blocking untagged cells from accessing newly arriving tokens. On the other hand, an arriving untagged cell will be buffered if the buffer is not full, otherwise it will be discarded. An untagged cell, reaching the head of the queue and finding the token pool empty, will wait until a new token arrives. It will then obtain the token and enter the network.

Comparing BLBP with BLB, one can find the results for the former by applying the analysis of multiplexer with priority scheme as presented in Section 3.4.4, with $K = K'+B$ and $K_l = K_l'+B$. However, with the additional condition that pretagged cells are also discarded if they reach the head of the queue and no token is available, the pretagged cells are expected to have a reduced chance of entering the network. This implies that the actual loss probability for pretagged traffic will be higher while the loss probability for untagged traffic will be lower than in the case without such a condition. Thus, we expect that the priority multiplexing analysis with $K = K' + B$ and $K_l = B$, where the state of empty token pool acts as an additional buffer threshold for pretagged cells, rather than the actual input buffer threshold, will give better approximation.

Following a similar derivation for the loss probability of multiplexing with priority, as in Section 3.4.4 with $K = K' + B$ and $K_l = B$, and depending on the state of queue, either cells of any kind can be accepted (if queue length j is not greater than K_l'), or only untagged cells can be accepted (if queue length j is greater than K_l'), we can define the following conditional probability mass function $\{a_2(i|j)\}, i = 0, 1, 2, \ldots,$ as

$$a_2(i|j) = \begin{cases} a_2(i) & \text{if } 0 \le j \le K_l \\ a_{2h}(i) & \text{if } K_l + 1 \le j \le K \end{cases} \tag{4.20}$$

where

$$a_{2h}(i) = (\eta\lambda_2)^i \exp(-(\eta\lambda_2))/i\,! \tag{4.21}$$

Following the same derivation as for (3.59) and (3.60), the cell loss probabilities for both untagged and pretagged cells can then be written as

$$P_{dh(BLBP)} = \sum_{k=1}^{\infty} s_2(k) \sum_{j=0}^{K} q_2^-(k,j)\, L_{2h}(k,j) \tag{4.22}$$

$$P_{dl(BLBP)} = \sum_{k=1}^{\infty} s_2(k) \sum_{j=0}^{K} q_2^-(k,j)\, L_{2l}(k,j) \tag{4.23}$$

where $L_{2h}(k,j)$ and $L_{2l}(k,j)$ are given by

$$L_{2h}(k,j) = \begin{cases} \sum_{i=K-j+2}^{\infty}[i-(K-j+1)]\,\eta\, a_2(i) & \text{if } 0 \le j \le K_l \\ \sum_{i=K-j+2}^{\infty}[i-(K-j+1)]\, a_{2h}(i) & \text{if } K_l + 1 \le j \le K \end{cases} \tag{4.24}$$

$$L_{2l}(k,j) = \begin{cases} \sum_{i=K-j+2}^{\infty}[i-(K-j+1)]\,(1-\eta)\, a_2(i) & \text{if } 0 \le j \le K_l \\ \sum_{i=1}^{\infty} i\, a_{2l}(i) & \text{if } K_l + 1 \le j \le K \end{cases} \tag{4.25}$$

and $a_{2l}(i) = ((1-\eta)\lambda_2)^i \exp(-((1-\eta)\lambda_2))/i\,!$

4.5.3 Buffered Dual Leaky Bucket with Priority (BDLBP)

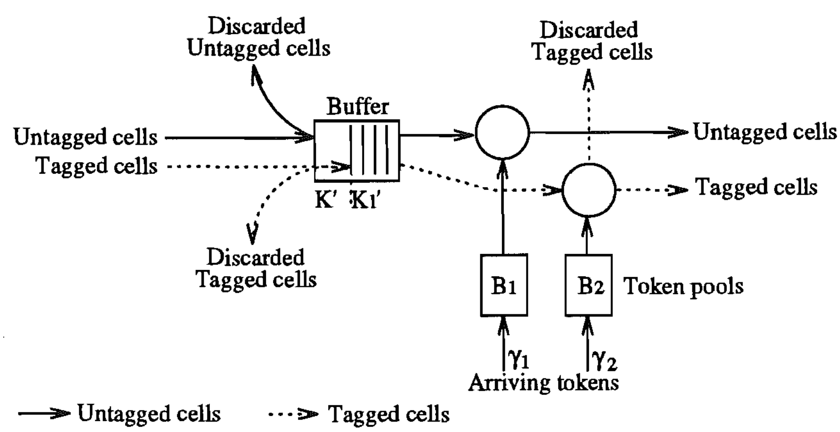


Figure 4.9 The buffered dual leaky bucket with priority (BDLBP).

Although the buffer threshold in BLBP limits the number of cells that can queue in the input buffer, the competition for tokens still exists once the tagged cells gain access into the input buffer and tokens are available. In order to completely prevent such competition for tokens, a possible modification to the scheme can be made through introducing an additional token pool for separately policing untagged and tagged traffic. The resulting scheme, called the *buffered dual leaky bucket with priority (BDLBP)*, is shown in Figure 4.9.

In this scheme, untagged and tagged cells are queued in a shared buffer, as opposed to separate buffers, in order to prevent cells from being delivered out of sequence. This shared buffer arrangement, however, leads to two potential problems. First, an untagged cell at the head of the queue blocks tagged cells from accessing available tokens in the second token pool, and second, the reverse situation where a tagged cell blocks untagged cells from accessing available tokens in the first token pool. The former situation results in the possibility of the queue being filled up very quickly by tagged cells causing newly arriving untagged cells to be discarded. In order to minimise this possibility, an additional buffer threshold K_l' is added to limit tagged cells. An arriving tagged cell will be buffered if the queue is less than K_l', otherwise it will be discarded. On the other hand, to resolve the latter problem, a condition is placed that a tagged cell will be discarded if it reaches the head of the queue and finds no available token in the second token pool. Overall, these two solutions suggest a scheme that will minimise the loss of untagged cells at the expense of the tagged cells. An arriving untagged cell will only be dropped if the buffer overflows, otherwise it will be buffered until it obtains a token from the first token pool and enters the network.

Because of the relative complexity of this scheme, let us use an approximation. Due to the abovementioned condition, we can make a simplifying assumption that tagged cells are not buffered at all, or equivalently, that the buffer is only accessed by the untagged cells. This basically decouples the leaky bucket into two separate buckets. The first bucket can be evaluated as an IPP/D/1/K queue with $K = K' + B_1$ and the untagged traffic parameters, $\lambda_2 = \eta\lambda$, $r_1 = \alpha$, and $r_2 = \beta$ being normalised to its token rate γ_1. The cell discarding rate, which corresponds to the loss probability for untagged or high priority cells, is given as

$$P_{dh(BDLBP)} = P_{IPP/D/1/K'+B_1} \tag{4.26}$$

The second bucket can be evaluated as an IPP/D/1/K queue with $K = B_2$ and the tagged traffic parameters, $\lambda_2 = (1-\eta)\lambda$, $r_1 = \alpha$, and $r_2 = \beta$ being normalised to its token rate γ_2. The cell discarding rate, which corresponds to the loss probability for tagged or low priority cells, is given as

$$P_{dl(BDLBP)} = P_{IPP/D/1/B_2} \tag{4.27}$$

The accuracy of the approximation was assessed by using simulation; see Section 4.6.

4.5.4 Buffered Dual Leaky Bucket with Priority and Dynamic Threshold (BDLBP-DT)

In BDLBP, described in the previous section, tagged cells may be queued behind an untagged cell, and when they reach the head of the queue (i.e. after the transmission of an untagged cell) and find no available token in the second token pool, they are discarded. This indicates a waste of buffer space, while at the same time untagged cells may be discarded due to the lack of buffer space. In order to overcome this drawback, a modification of the BDLBP by shifting the second bucket to the front of the input buffer suggests itself. The modified scheme, shown in Figure 4.10, is referred to as the *buffered dual leaky bucket with priority and dynamic threshold (BDLBP-DT)*.

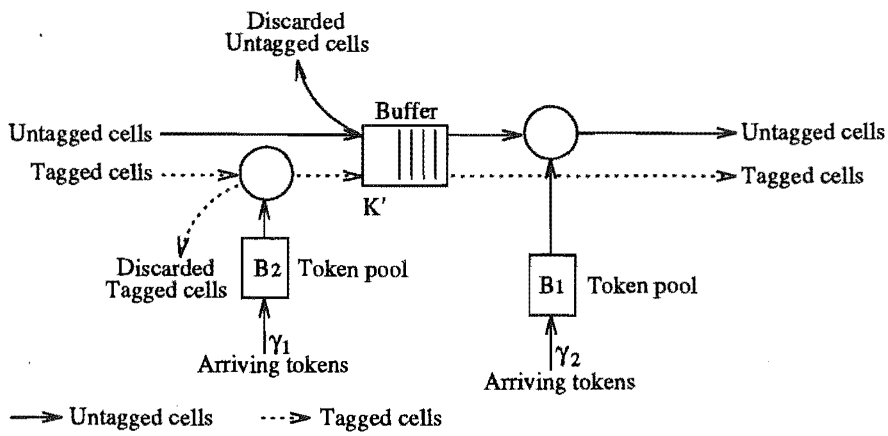


Figure 4.10 The buffered dual leaky bucket with priority and dynamic threshold (BDLBP-DT).

Here, an arriving tagged cell has to possess a token before joining the queue, provided that the buffer is not full. If the buffer is full or if no token is available, then the cell will be discarded. The number of tagged cells in the queue is limited by the bucket depth of the second token pool, which acts as if a dynamic buffer threshold. When the tagged cell, which gains access into the queue, reaches the head of the queue, it will be transmitted. On the other hand, an arriving untagged cells will enter the queue and wait in the queue for an available token in the first token pool. If the buffer is full, then the arriving cell will be discarded.

As described above, the fate of tagged cells depends on the queue length of the input buffer. The queued tagged cells will in turn affect the access of other tagged cells to the queue. Such interdependency of tagged cells on the queue length prevents us from deriving an exact analytical solution for the scheme, therefore we use an approximation. We assume that the tagged cells can access tokens from the second bucket independently of the queue length of the input buffer and that they are not queued in the input buffer. In this sense, the BDLBP-DT can be viewed as two separate leaky buckets, resulting in a similar solution as in the case of BDLBP, i.e.

$$P_{dh(BDLBP-DT)} = P_{dh(BDLBP)} = P_{IPP/D/1/K'+B_1} \tag{4.28}$$

$$P_{dl(BDLBP-DT)} = P_{dl(BDLBP)} = P_{IPP/D/1/B_2} \tag{4.29}$$

The accuracy of the approximation was assessed using simulation; see Section 4.6.

4.5.5 Dual Leaky Bucket with Marking and Priority (DLBMP)

The *dual leaky bucket with marking and priority (DLBMP)* was proposed, but not analysed, by Eckberg [1992]. It can be considered as a modification of the DLBM in the situation where a source generates both tagged and untagged traffic. At the input of the leaky bucket, arriving cells are separated and only untagged cells are policed by the first bucket, and marked if no token is available. The tagged cells combined with the marked cells are policed by the second bucket. Any marked or tagged cells that find no tokens available in the bucket will be discarded. This implies that the cell loss at the leaky bucket will be originated only from tagged cells and from untagged cells that have been marked.

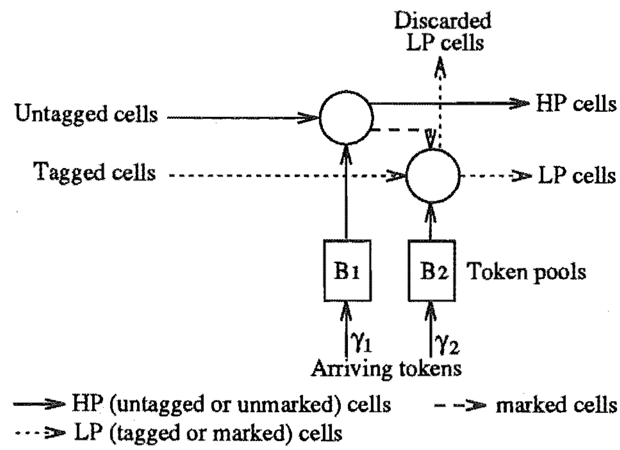


Figure 4.11 The dual leaky bucket with marking and priority (DLBMP).

Without the marking scheme, DLBMP is equivalent to DLBP with $\lambda_h = \eta\lambda$ and $\lambda_l = (1-\eta)\lambda$ being offered to the first and the second bucket, respectively. Such inclusion of marking increases by $\lambda_h P_{m(DLBMP)}$ the amount of traffic to the second bucket, where the probability $P_{m(DLBMP)}$ of untagged cells being marked by the first bucket is equal to the cell discard rate $P_{dh(DLB)}$ of DLB, i.e.

$$P_{m(DLBMP)} = P_{dh(DLB)} = P_{IPP/D/1/B_1}, \tag{4.30}$$

with the IPP parameters $\lambda_2 = \lambda_h$, $r_1 = \alpha$, and $r_2 = \beta$ being normalised to its token rate γ_1.

The total amount of traffic to be policed by the second bucket is equal to $\lambda_2 = \lambda_l + \lambda_h P_{m(DLBMP)}$. The discard rate of this traffic depends on the second bucket, which can be evaluated by using the IPP/D/1/K with $K = B_2$ and the IPP parameters being normalised to the token rate γ_2. The discard rate is

$$P_{d2} = P_{IPP/D/1/B_2} \tag{4.31}$$

The discard rate of the tagged cells is simply equal to P_{d2}, whereas the loss probability for untagged cells must be conditioned upon the fact that they are marked first. Overall these loss probabilities can be expressed as

$$P_{dh(DLBMP)} = P_{d2} P_{m(DLBMP)} \tag{4.32}$$

$$P_{dl(DLBMP)} = P_{d2} \tag{4.33}$$

4.5.6 Buffered Dual Leaky Bucket with Marking and Priority (BDLBMP)

The *buffered dual leaky bucket with marking and priority (BDLBMP)* can be considered as an adaptation of a BDLBM for policing sources with pretagged cells, or as a modification of a BDLBP with the addition of marking. The operation of the scheme is similar to that of BDLBP, except that instead of discarding a newly arriving untagged cell when the buffer is full, the cell at the head of the queue is marked if it is an untagged cell or discarded if it is a tagged cell to make space for the newly arrived cell. The aggregate of low priority cells, which comprise tagged and marked cells, are policed by the second bucket, hence limiting the number of low priority cells entering the network.

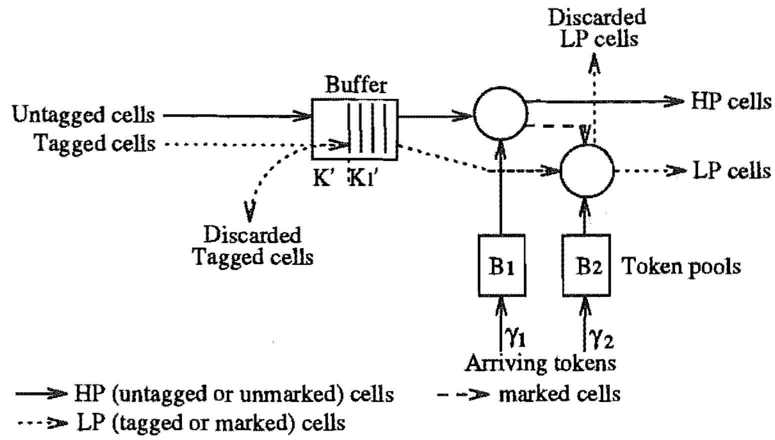


Figure 4.12 The buffered dual leaky bucket with marking and priority (BDLBMP).

An exact analysis of this scheme does not seem to be possible. Hence we use a similar approximation as in the case of BDLBP, where the probability of untagged cells being marked by the first token pool was assumed to be equal to the cell discarding rate of the first token pool in the BDLBP, i.e.

$$P_{m(BDLBMP)} = P_{d(BDLBP)} = P_{IPP/D/1/K'+B_1} \tag{4.34}$$

These marked cells contribute to the low priority traffic that is policed by the second token pool with $\lambda_2 = (1-\eta)\lambda + \eta\lambda P_{m(BDLBMP)}$. The discard rate at the second bucket P_{d2} can be evaluated by using the IPP/D/1/K with $K = B_2$, i.e.

$$P_{d2} = P_{IPP/D/1/B_2} \tag{4.35}$$

Just like DLBMP, the discard rate of the tagged cells is simply equal to P_{d2}, whereas the loss probability for untagged cells must be conditioned upon the fact that they are marked first. Overall these loss probabilities can be expressed as

$$P_{dh(BDLBMP)} = P_{d2}P_{m(DLBMP)} \tag{4.36}$$

$$P_{dl(BDLBMP)} = P_{d2} \tag{4.37}$$

The accuracy of the approximation will be assessed by using simulation in Section 4.6.

4.6 Verification of Analytical Results

In this section, we will verify the analytical results against simulation results for varying bucket depths, token rates and buffer sizes. Simulation is carried out by using DESC++ [Hartanto, 1993]. We simulate a single video source generating both high and low priority cells with parameters given in Table 3.2. The ratio of high priority traffic intensity to the overall traffic is chosen to be 0.5. We fed the traffic from the source through various leaky bucket schemes and obtain the cell discard rate. All simulation results are required to have 0.05 precision at a 95% confidence interval.

4.6.1 Non-Priority Schemes

In our studies of single bucket schemes, we choose $B = 1700$, $\gamma = p\lambda$ or equivalently $f_R = 1.0$, and $K' = 50$. The bucket depth is selected to be lower than the average burst length of the source which is approximately equal to 3416, as it is still possible to analyse despite exponential growth of state space when solving the schemes analytically. For dual bucket schemes, e.g. DLBM and BDLBM, we choose the bucket depth and the token rate of the second bucket to be 0.25 times the bucket depth and the token rate of the first bucket, respectively. In the following graphs, only total cell loss probability is shown for each scheme, because these schemes do not differentiate between untagged and tagged cells from the users.

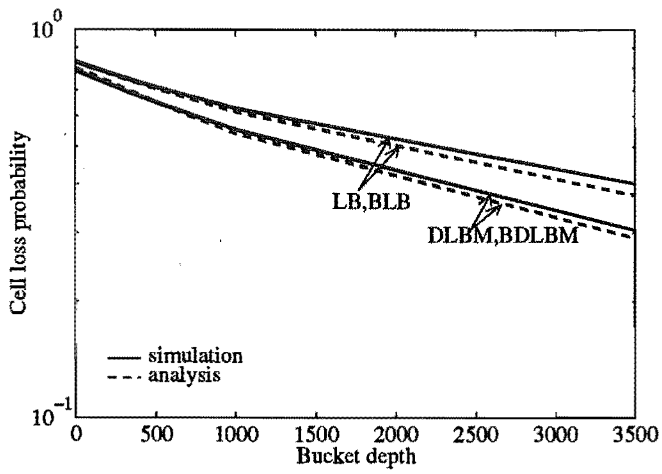


Figure 4.13 Verification of analytical results of non-priority schemes for various bucket depths.

Figure 4.13 shows the cell discard rate for various bucket depths under LB, BLB, DLBM, and BDLBM. One can see that the analytical results are very close to the simulation results. They show good agreement especially for small bucket depth values, where the difference is less than 1%. This agreement sligthly degrades as the bucket depth increases, which may be due to the limit on the state space in computing the analytical results. The results also justify the approximation used in obtaining results for DLBM and BDLBM.

Figure 4.14 shows the cell discard rate for various normalised token rates (f_R). Again the graph demonstrates a good agreement between the analytical and simulation results, where the difference in the results is less than 1% at $f_R = 1.0$. In general, the results degrade as token rate increases, which again may be due to the limitation imposed on the state space required in the case of analytical studies. However, the difference in the results is still below 10%, even at $f_R = 3.0$.

Figure 4.15 shows the cell discard rate for various buffer sizes. The analytical results of BLB and BDLBM schemes show good agreements with the simulation results for all buffer sizes, where the difference is less than 5%.

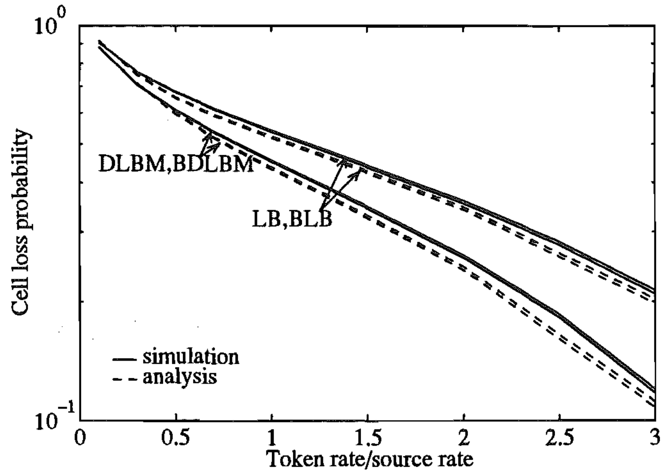


Figure 4.14 Verification of analytical results of non-priority schemes for various normalised token rates (f_R).

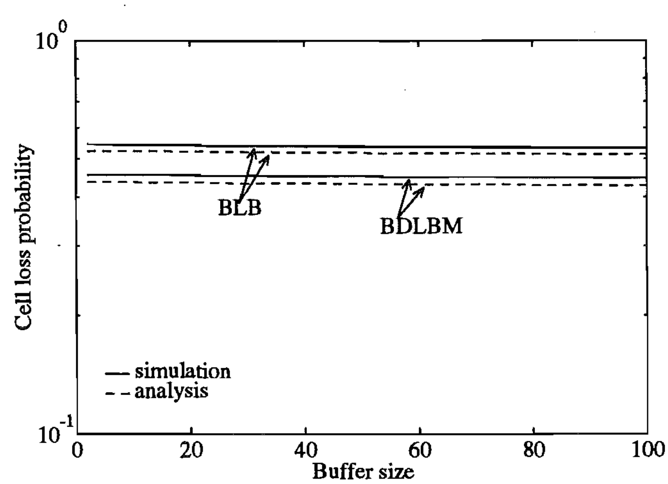


Figure 4.15 Verification of analytical results of non-priority schemes for various buffer sizes.

4.6.2 Priority Schemes

For studying leaky bucket schemes serving streams of cells of different loss priorities we have selected the same working condition as for the single bucket schemes of non-priority case, i.e. $B = 1700$, $\gamma = p\lambda$ or equivalently $f_R = 1.0$, and $K' = 50$. For dual bucket schemes, we choose the bucket depths and the token rates of the first and the second bucket such that $B_1 + B_2 = B$ and $\gamma_1 + \gamma_2 = \gamma$. In order to show the preferential treatment for the untagged traffic, we choose

B_1 and γ_1 to be three times larger than B_2 and γ_2, i.e., $B_1 = 0.75B$ and $\gamma_1 = 0.75\gamma$. For the schemes with additional buffer threshold, e.g. BLBP, we set the buffer threshold at 0.25 of the overall buffer size, i.e. $K_l' = 0.25K'$.

In Figure 4.16 we plot the loss probabilities for high and low priority cells for various priority schemes obtained from analytical and simulation studies. The results for DLBMP and BDLBMP schemes show the loss probabilities for the *original* loss priority (i.e. the loss priority indicated by the users) of the discarded cells.

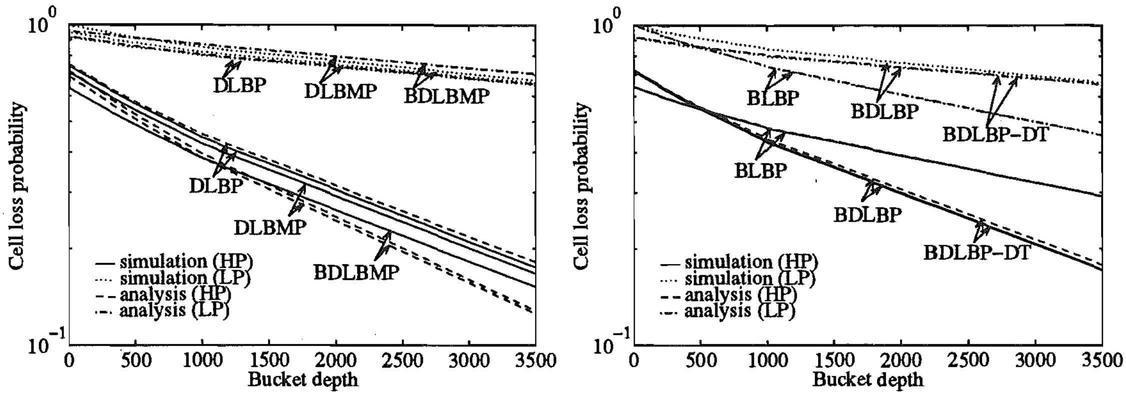
The graph shows an excellent agreement between the analytical and simulation results for low priority cell loss probability for DLBP, BLBP and BDLBP-DT, over the entire range of bucket depths, where the difference is less than 1%. The approximation used in the analysis of DLBMP and BDLBMP seems justified as the results show less than 10% difference, although the accuracy of the approximation degrades as bucket depth increases. Again this may be caused by the limited state space problem. One interesting observation on the plot for BDLBP scheme, is that the agreement of the results improves as the bucket depth increases. The large deviation in the case of low bucket depth may be explained by the fact that in the analytical model, it was assumed that low priority cells are not queued in the input buffer. With low bucket depth, we can expect that high priority cells will also be queued due to unavailability of tokens. This results in large queueing and hence the queue in the buffer tends to grow beyond K_l. With increasing bucket depth, this queueing reduces and hence the chance of low priority cells being discarded due to the buffer threshold is also reduced.

The graph also shows a good approximation of loss probability for untagged cells in the case of DLBP, BLBP, BDLBP and BDLBP-PT, where the difference is less than 5%. On the other hand, the quality of results for DLBMP and BDLBMP degrades substantially with the increasing of bucket depths.

Figure 4.17 shows the cell loss probabilities for untagged and tagged cells for varying normalised token rates (f_R). In general, the figure shows a good agreement between analytical and simulation results for all priority schemes. The agreement between simulation and (approximated) analytical results degrades as the token rate increases, from the relative difference of 3% at $f_R = 0.3$ to 28% at $f_R = 3.0$.

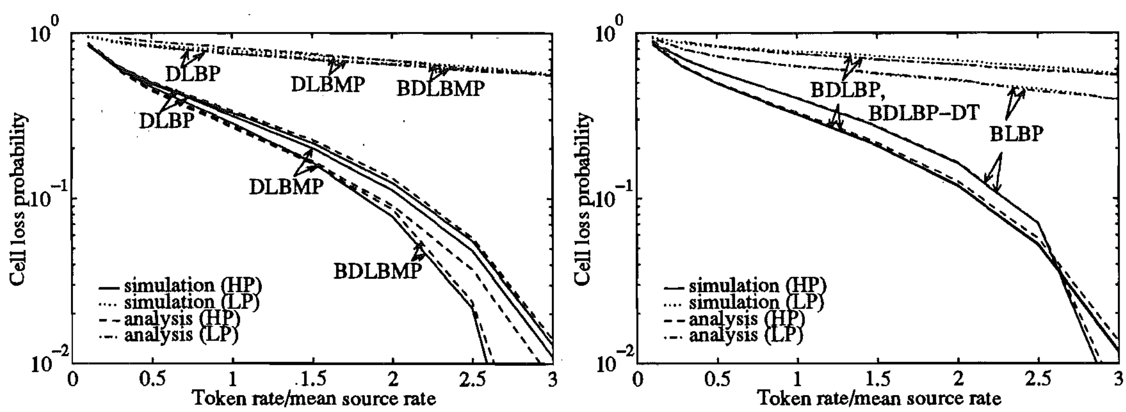
Figure 4.18 shows good quality of analytical approximations of the loss probabilities for untagged and tagged traffic as compared with the results obtained from simulation for varying buffer size. As expected for all schemes, the loss probabilities for high priority cells obtained from analytical approximation are higher than the simulation ones. This is due to the fact that the interference between high and low priority streams is not taken into account in the analytical model.

In summary, the analytical results have been shown to provide reasonable agreements to the simulation ones, where the difference is less than 10% for most cases. These results are useful in studying the performance of the leaky bucket individually. However, since approximation has been used for some of the schemes, e.g. BDLB and BDLB-DT which results in equal cell loss probabilities between the schemes, we could not compare the performance of the schemes. For this reason, simulation will be used in the next section for comparing the performance of the schemes.



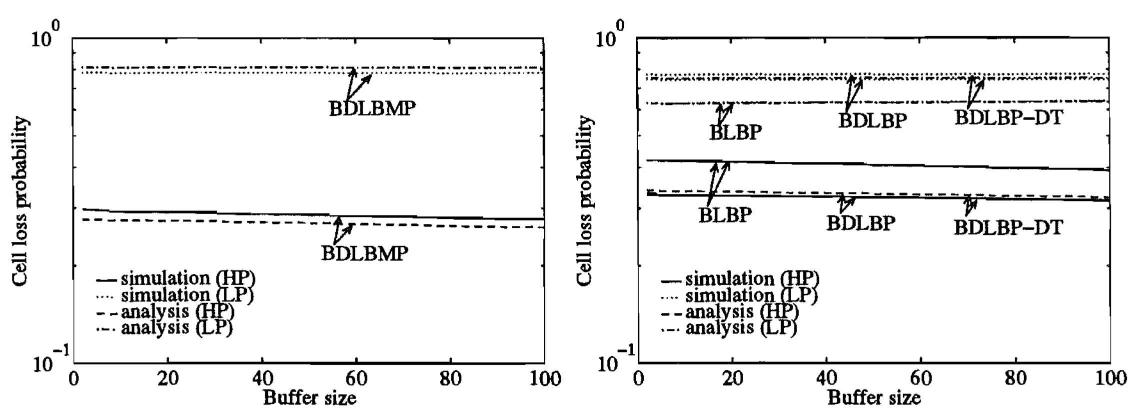
(a) DLBP, DLBMP, BDLBMP (b) BLBP, BDLBP, BDLBP-DT

Figure 4.16 Verification of analytical results of priority schemes for various bucket depths.



(a) DLBP, DLBMP, BDLBMP (b) BLBP, BDLBP, BDLBP-DT

Figure 4.17 Verification of analytical results of priority schemes for various normalised token rates (f_R).



(a) BDLBMP (b) BLBP, BDLBP, BDLBP-DT

Figure 4.18 Verification of analytical results of priority schemes for various buffer sizes.

4.7 Performance Comparisons

In this section, we compare the performance of leaky buckets used for policing the average rate of a source as well as the performance of a multiplexer fed by policed sources and the quality of service experienced by a source. The measure of performance, which is of primary interest here, is the loss probability of untagged cells. A leaky bucket is considered better than another if fewer cells are lost end-to-end whilst minimising the untagged cells it discards.

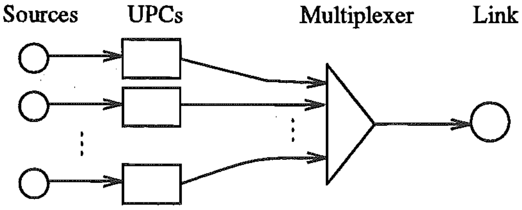


Figure 4.19 The simulation model.

The simulated configuration consists of N homogeneous IPP sources feeding traffic into a multiplexer as shown in Figure 4.19. The sources are policed by one of the leaky bucket schemes described in the previous sections. The parameters of IPP sources are taken from the video traffic parameters listed in Table 3.1. We choose $N = 10$ and the fraction of high priority traffic $\eta = 0.5$, which represents a typical upper bound value; see for example a voice source using the coding scheme proposed by Sriram *et al.* [1991].

As the reference, we take the bucket depth and the token rate of the leaky buckets to be equal to the average burst length ($B = 3416$) and the mean rate of the source, respectively. This implies that the ratio f_R between the token rate and the mean traffic rate of the source is equal to 1.0. In non-priority schemes which have two token pools, e.g. in BDLBM, we limit the marked traffic rate by choosing $\gamma_2 = 0.25\gamma_1$ and $B_2 = 0.25B_1$. In priority schemes which have two token pools, e.g. in BDLBP, we can control the ratio of bucket depths and token rates for the same total bucket depth and token rate as those of single token pool. In order to enhance the performance of high priority traffic, we will set the ratio to be 3:1. In other words, we choose the *split factor* f_S, which was defined in Section 4.3 as the ratio between the token rate and bucket depth of the first token pool in a leaky bucket of the class of dual bucket with priority to the total token rate and bucket depth, to be 0.25. The buffer size for the leaky bucket schemes employing buffering, e.g. BLB, is chosen to be 50, i.e. $K' = 50$ cells. The buffer threshold in BLBP, BDLBP and BDLBMP is chosen to be 0.25 of the buffer size, i.e. $K'_l = 0.25K'$.

The multiplexer is assumed to use a partial buffer sharing scheme with the buffer thresholds $K = 20$ and $K_l = 10$. Assuming a normalised token rate $f_R = 1.0$ implies that the total traffic offered to the multiplexer is equal to 1.0.

In the following sections, we only study the effects of varying the leaky bucket parameters and the source parameters on the cell loss probabilities at leaky buckets, at the multiplexer and on an end-to-end basis (i.e. the total of cells lost at a leaky bucket and at the multiplexer). The variation of multiplexer parameters is not considered as they only affect the performance of the multiplexer and to some extent the performance of the leaky bucket with marking schemes, such as LBM, BLBM, which is not of primary interest here.

4.7.1 Influence of Leaky Bucket Parameters

In presenting the results, we will differentiate the plots for non-priority class and priority class of leaky bucket schemes. Discussing the results, we will focus more on the quality of service offered to the untagged traffic than to the tagged traffic.

Variation of Bucket Depth

(a) Cell Loss at the Leaky Bucket

In order to compare the performance of various leaky bucket schemes, we plot the loss probability over a large range of bucket depths, see Figure 4.20(a). One can see that the additional buffer in BLB hardly affects the performance of the leaky bucket due to the bursty nature of the traffic. This is in contrast to the conclusion obtained from analysis of less bursty traffic in [Chao, 1991], where buffering as well as marking policy offers quite significant improvement.

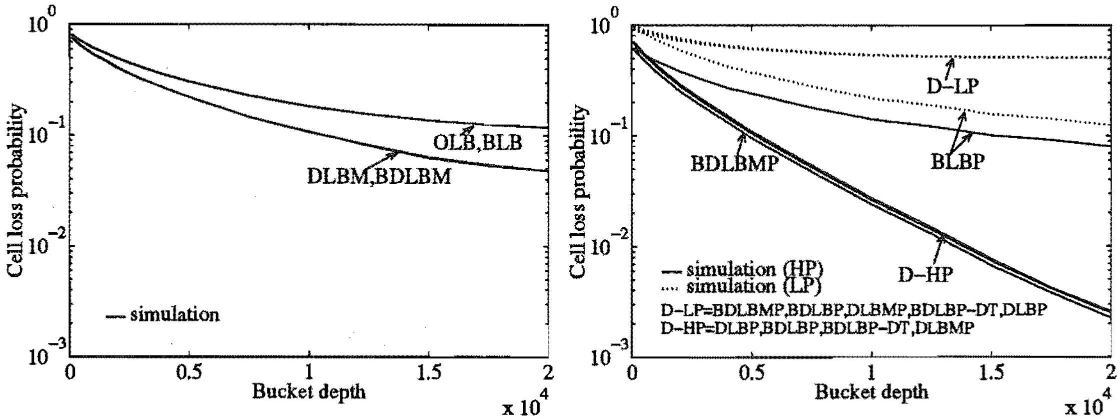
The results obtained for all priority schemes, except BLBP, seem to be qualitatively the same, see Figure 4.20(b). One can see the performance for high priority traffic, represented by D-HP, does not differ much from scheme to scheme. Better performance under BDLBMP is attributed to the buffer threshold for tagged traffic K_l^t and the marking scheme applied in the leaky buckets. Compared with DLBM, the buffer threshold reduces the number of tagged cells competing for tokens in the second bucket, while in comparison with BDLB, the marking scheme allows untagged cells to make use of tokens in the second bucket. In the last case the untagged cells face less competition due to the fact that tagged cells have already been dropped at the input buffer.

Comparing the performance of BDLBP and BDLBP-DT, the latter offers better performance through a better admission control of the tagged cells into the queue. This improvement is strongly dependent on the token rate and the bucket depth of the second bucket. As expected the performance of BDLBP-DT degrades with increasing values of these parameters, as more tagged cells were allowed to enter the queue and to compete for the input buffer space with the untagged cells. In such a situation, the performance of BDLBP-DT can be worse.

Comparing the performance for the leaky buckets with non-priority and priority schemes, we can see that all priority schemes offer much better quality of service for untagged traffic than their non-priority counterparts. As shown in the graphs, the loss probabilities for untagged cells in the priority schemes are more than one order of magnitude lower than that in the non-priority ones. However, the improvement in the case of BLBP as compared to OLB and BLB schemes is not as much as that observed by Waterman *et al.* [1993] for less bursty traffic.

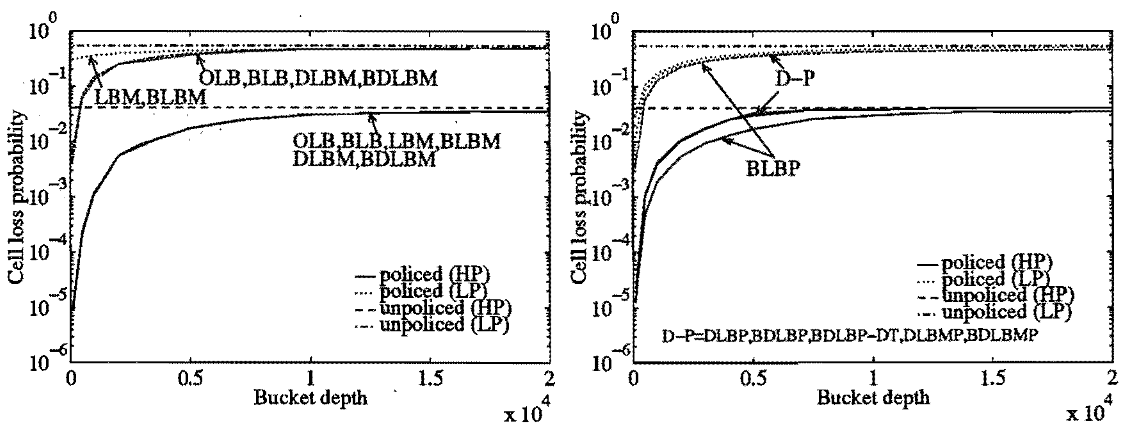
(b) Cell Loss at the Multiplexer

In Figure 4.21(a), we show the performance of a multiplexer fed by the policed sources. In conjunction with Figure 4.20, the results show that the improvement in the cell loss probability at the UPC achieved by increasing the bucket depths has come at the expense of increasing the cell loss probability at the multiplexer as more bursty traffic is allowed to enter the network. This trend has been observed previously by Sidi *et al.* [1989] and Hughes *et al.* [1990b] for Poisson and Bernoulli traffic input, respectively. Comparing the performance of policed sources



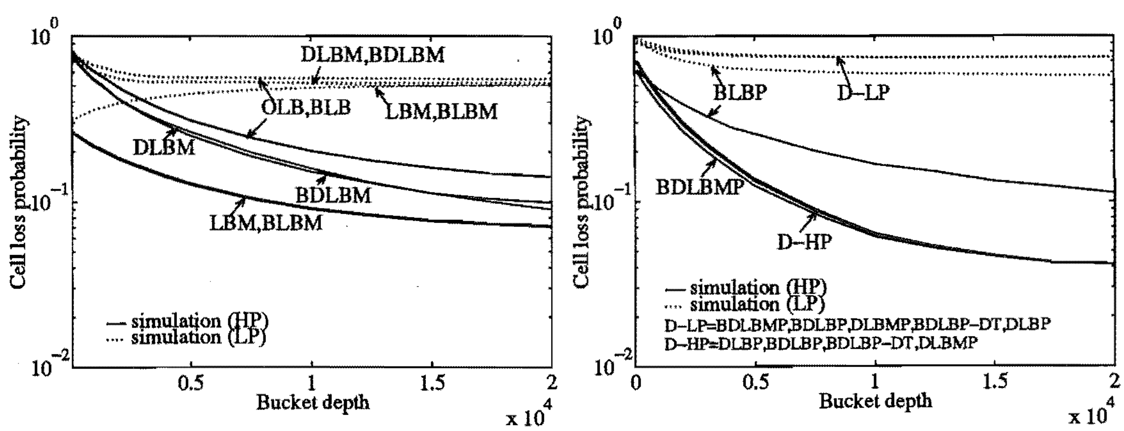
(a) Non-priority schemes (b) Priority schemes

Figure 4.20 Cell loss probability at the UPC versus bucket depths for $f_R = 1.0$.



(a) Non-priority schemes (b) Priority schemes

Figure 4.21 Cell loss probability at the multiplexer versus bucket depths for $f_R = 1.0$.



(a) Non-priority schemes (b) Priority schemes

Figure 4.22 End-to-end cell loss probability versus bucket depths for $f_R = 1.0$.

with unpoliced sources, we can see that the more stringent the policer, the better the performance quality offered by the multiplexer. The policing effect diminishes for very large bucket depths. The marking schemes, LBM and BLBM, which apply no cell discarding at the leaky bucket, experience higher cell loss at the multiplexer. Although apparently the high priority cell loss probabilities for these schemes are much lower than the ones for low priority traffic, the actual loss probabilities by considering the original cell loss priority, are much higher. Simulation results reveal that on average between 6-40% of the discarded low priority cells are actually marked cells, which were originally of high priority. The smaller the bucket depth, the higher the number of high priority cells which are marked and the higher also the actual cell loss probability of the high priority cells. DLBM and BDLBM schemes, which limit the low priority traffic, have also lower loss probability of the traffic at the multiplexer. From the priority schemes considered, see Figure 4.21(b), BLBP offers the best performance for the high priority traffic.

(c) End-to-End Cell Loss Probability

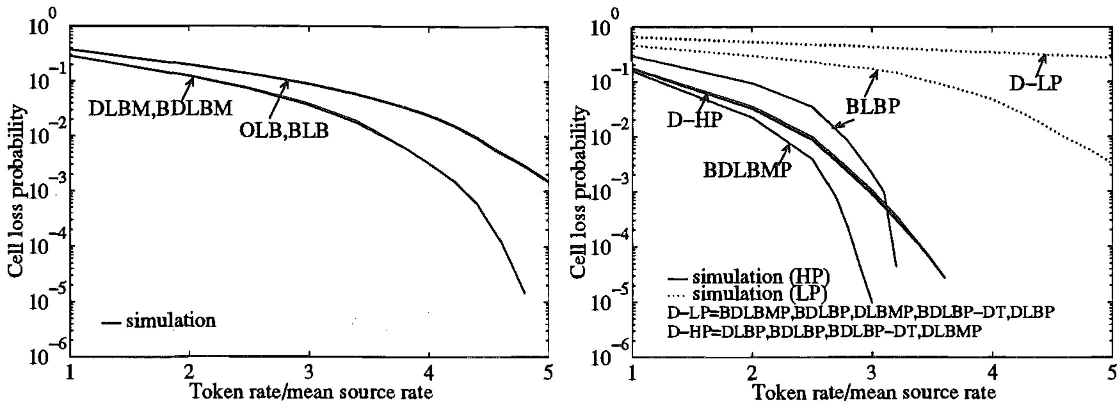
The actual cell loss probability experienced by a source can be better examined on the basis of the end-to-end performance, where the cell loss probability is measured according to the loss priority of a cell before it was policed by a leaky bucket. This implies that the loss of high priority cells in a marking scheme comprises the loss of untagged cells, regardless whether they have or have not been marked by the leaky bucket.

We plot in Figure 4.22(a) the end-to-end performance for non-priority schemes. The graphs show that one can secure better performance through marking cells (LBM and BLBM) than shaping traffic (BLB). However, since marked cells may be discarded indiscriminately by the multiplexer, there is no guarantee on the delivery of these cells. The graphs also show that the end-to-end performance quality from the point of view of high priority cells is much better under LBM and BLBM than that shown in Figure 4.21(a). As expected, the performance of LBM and BLBM is better than DLBM and BDLBM, where the marked traffic rate is limited. Overall, the schemes offer better performance than that under OLB and BLB, both for untagged and tagged cells. Figure 4.22(b) demonstrates that an even better performance can be achieved using the schemes that consider pretagged traffic. Among them, BDLBMP offers the best performance for untagged traffic without sacrificing too much the quality of service offered to tagged cells. Higher loss probability of tagged cells is due to the fact that the cells have to compete with marked cells for the tokens of the second bucket.

Variation of Token Rate

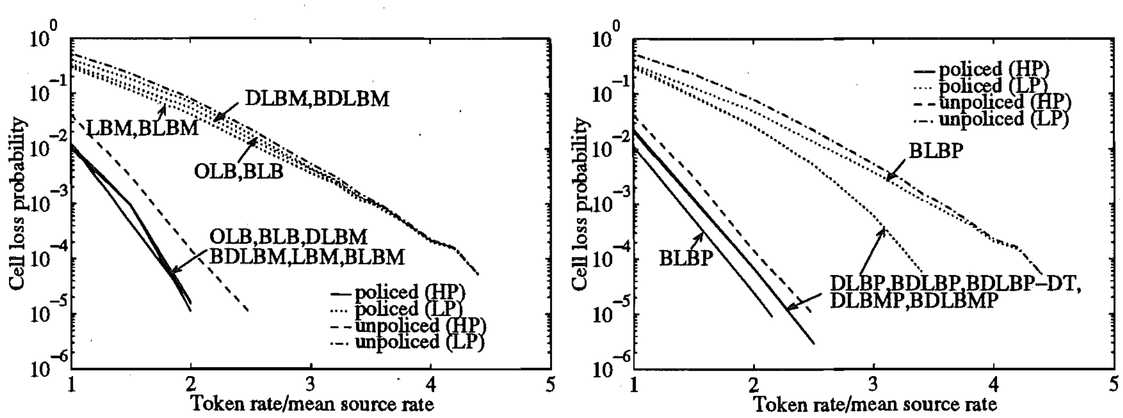
(a) Cell Loss at the Leaky Bucket

In Figure 4.23(a), we compare the performance of the leaky bucket schemes for varying *normalised token rate* (f_R), which is defined in Section 4.3 as the ratio between the token rate to the source mean rate. We observe that most leaky bucket schemes studied here required a very large f_R in order to achieve a typical required value of cell loss probability for untagged traffic of about 10^{-9}. OLB and BLB can not be expected to satisfy the required level of cell loss probability without increasing the bucket depth.



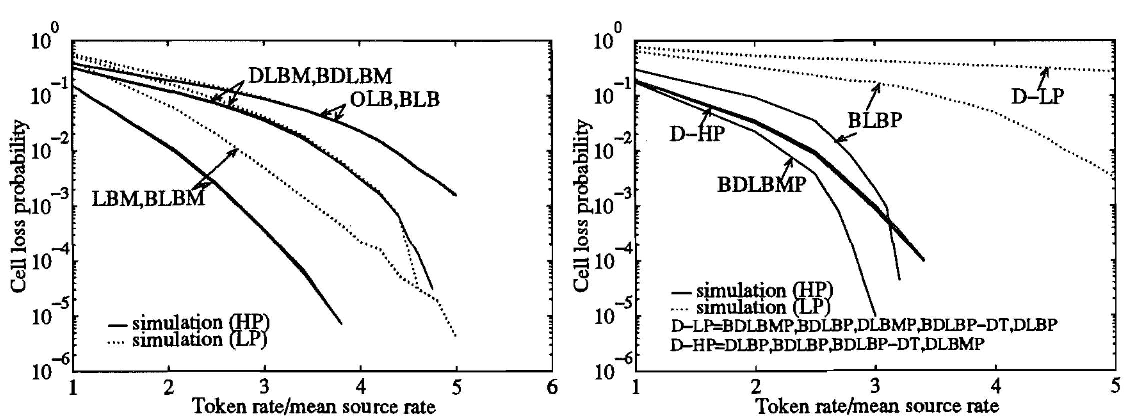
(a) Non-priority schemes (b) Priority schemes

Figure 4.23 Cell loss probability at the UPC versus normalised token rate for $B = 3416$.



(a) Non-priority schemes (b) Priority schemes

Figure 4.24 Cell loss probability at the multiplexer versus normalised token rate for $B = 3416$.



(a) Non-priority schemes (b) Priority schemes

Figure 4.25 End-to-end cell loss probability versus normalised token rate for $B = 3416$.

One can see in Figure 4.23(b) that BDLBMP gives the best performance among the considered priority schemes. On the other hand, BLBP gives the worst performance when $f_R < 3.1$, but for larger value of f_R, it gives better performance than the rest of the dual leaky bucket schemes, except BDLBMP. The sharp improvement in the loss probability at $f_R \approx 3.0$ is due to the sharing of token pool by both tagged and untagged traffic. Such sharing causes free competition between untagged and tagged cells which have gained access into the input buffer, when the availability of tokens is rare, i.e. for small f_R. When we increase f_R, the number of available tokens is increased. Since the number of tagged cells which gain access into the input buffer stays the same, this implies that the share of tokens used by the untagged cells is greater and hence the probability of arriving untagged cells finding empty token pool is reduced, and so the loss probability of untagged trafffic decreases.

In the case of dual bucket schemes, the reduction in loss probabilities for untagged traffic with increasing token rate is slower. This is due to the separate reservations of tokens for tagged and untagged traffic, where free tokens reserved for tagged cells can not be used by untagged cells as in the case of DLBP. On the other hand, competition for tokens in the second token pool exists between marked and tagged cells under DLBMP even for large f_R value.

Comparing Figures 4.23(a) and (b), one can see that the priority schemes allow to achieve the required cell loss level at, say 10^{-9}, using much smaller normalised token rate. Since the token rate can be interpreted as the bandwidth allocated for each connection, one can conclude that the schemes offer better resource utilisation than non-priority schemes.

(b) Cell Loss at the Multiplexer

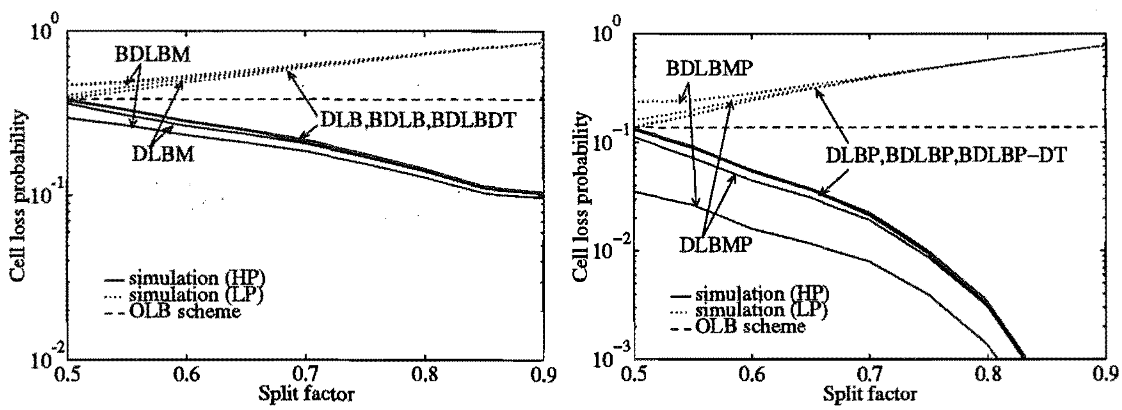
As shown in Figure 4.24, we see that the cell loss probability at the multiplexer for policed sources is smaller than that for unpoliced ones, especially when the normalised token rate is smaller than 1.0. This is due to the high cell discarding rate at the leaky bucket, which results in a very small number of cells that actually gets through. Hence, less competition for the buffer space at the multiplexer occurs. On the other hand, under the marking scheme a lot of high priority cells are marked, resulting in high competition among low priority cells for the multiplexer buffer and high cell loss. Since the total bandwidth equals the sum of token rates, we can say that increasing the normalised token rate will increase the amount of bandwidth to be shared among the sources. Thus, the performance of the multiplexer improves. The same trend can be observed for the priority schemes; see Figure 4.24(b).

(c) End-to-End Cell Loss Probability

From the dependence of the end-to-end cell loss probability on the normalised token rate (see Figure 4.25(a)), one can see that the same QoS is offered for tagged and untagged cells under OLB, BLB, DLBM, and BDLBM. This is primarily due to the fact that more cells are discarded at the leaky bucket. With no differentiation of the traffic at the leaky bucket, this loss probability will be the same for both types of traffic. Under LBM and BLBM, cell loss occurs within the network only. The better quality of service offered for the untagged traffic than for the tagged traffic is due to the different treatment provided by the partial buffer sharing scheme used by the multiplexer. The end-to-end cell loss probabilities under the priority schemes, see Figure 4.25(b), are also very similar to those at the UPC, since there are very small losses at the multiplexer within the network.

Variation of Split Factor

Comparing the results depicted in Figures 4.25(a) and (b), one can see quite significant improvement in the loss probabilities of high priority traffic due to the introduction of priorities. For example, when $f_R = 3.0$, the performance of the best priority scheme (i.e. BDLBMP) is approximately three orders of magnitude better than the best non-priority scheme (i.e. BDLBM). Such improvement is very much dependent on the *split factor* f_S, which is defined in Section 4.3 as the ratio between the token rate and the bucket depth of the first token pool to the total token rate and bucket depth. The results presented in Figure 4.25(b) were obtained for $f_S = 0.75$. The effects on the cell loss probabilities under priority schemes, due to variation in the split factor for $f_R = 1.0, f_B = 1.0$, and $f_R = 2.5, f_B = 1.0$, are shown in Figure 4.26(a) and (b), respectively. For comparison, the cell loss probabilities with those under OLB are also shown.



(a) $B = 3416$ and $f_R = 1.0$

(b) $B = 3416$ and $f_R = 2.5$

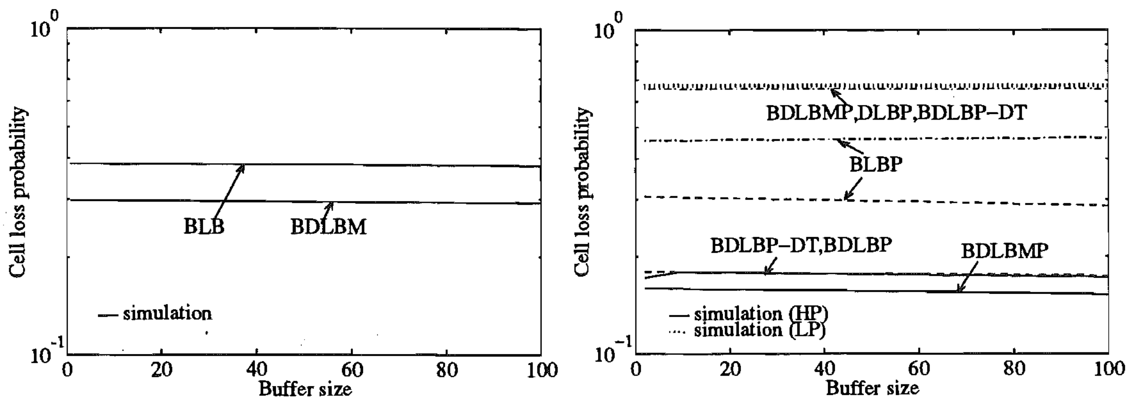
Figure 4.26 Cell loss probability at the UPC for dual bucket schemes versus split factor (f_S).

From the graphs, we see that for $f_S = 0.5$, the performance of the priority schemes, except DLBMP and BDLBMP, is similar to that observed under OLB; the better performance under DLBMP and BDLBMP is due to the fact that marked high priority cells are able to make use of some tokens in the second pool. The performance of the priority schemes improves as we increase f_S, hence reserving more tokens for untagged traffic than for tagged traffic. Coupled with the increase in the normalised token rate, the priority schemes can offer very substantial improvement of service quality for the untagged traffic.

Variation of Buffer Size

Figure 4.27 shows a very small reduction in the cell loss probability due to the variation of input buffer size. The analysis of results obtained for BLBP leads to different conclusions than those that could be formulated when voice sources are used. In the latter case the variation in buffer size significantly affects the results [Waterman *et al.*, 1993].

The variation of the buffer threshold for tagged traffic K_l^t in BDLB and BDLBM has also been



(a) Non-priority schemes

(b) Priority schemes

Figure 4.27 Cell loss probability at the UPC versus buffer sizes for $B = 3416$ and $f_R = 1.0$.

considered and it is shown to have a little effect on the leaky bucket performance. The results also show that the performance of BDLBM degrades below the performance of DLBM for $K_l' = 0$. This is due to the free competition between the untagged and tagged cells for the input buffer space, resulting in high losses of untagged cells.

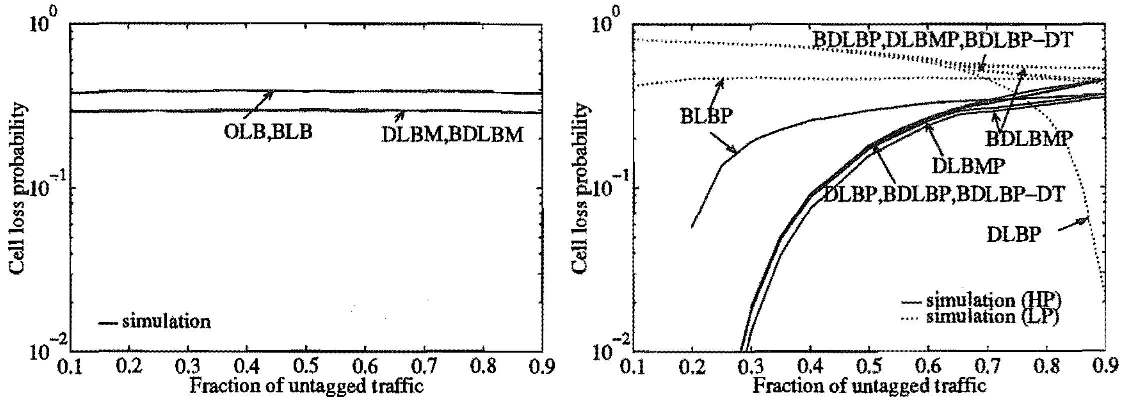
4.7.2 Influence of Source Parameters

Variation of Untagged Traffic Ratio

(a) Cell Loss at the Leaky Bucket

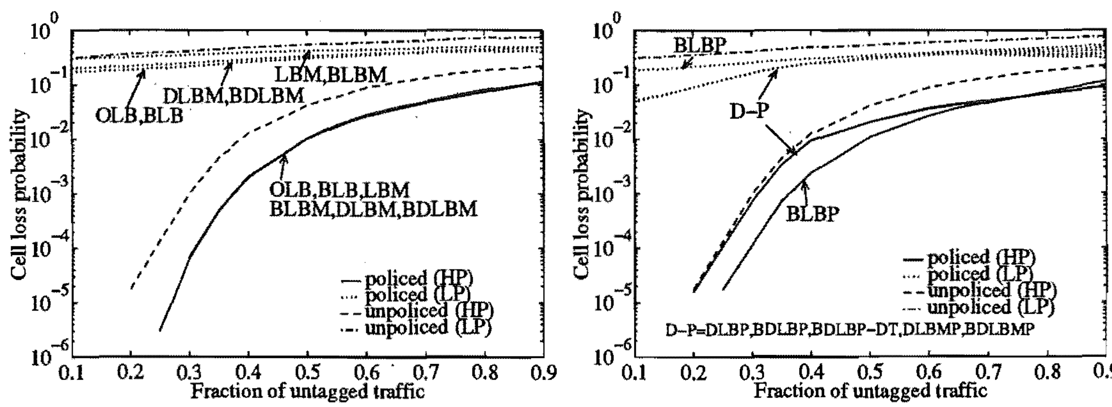
The effects that the fraction of untagged traffic η have on the performance of non-priority schemes are shown in Figure 4.28(a). As expected, the variation has no effect on the performance quality of the schemes. For priority schemes, the variation has a strong effect, see Figure 4.28(b). As we increase the ratio η, the cell loss probability of untagged traffic increases while the cell loss probability for tagged traffic decreases. One peculiar observation is that for $\eta > 0.75$, the loss probability of the tagged traffic under DLBP is lower than the loss probability for the untagged traffic. This anomaly is due to the token reservation controlled by the split factor f_S. By setting $f_S = 0.75$, we fix the token rates for the first and the second token pools. This means that the token arrival rate for the second bucket is $(1 - f_S)f_R = (1 - f_S)p\lambda$. As the tagged cell arrival rate is given by $(1 - \eta)p\lambda$, we can see that for $(1 - f_S) > (1 - \eta)$, tokens arrive at the second bucket faster than the tagged cells. Under DLBP, these extra tokens are not competed by untagged traffic, therefore arriving tagged cells will have greater chances of finding free tokens and so the cell loss probability for tagged traffic decreases.

The abundance of tokens in the second token pool implies a shortage of tokens in the first bucket, because the total number of tokens is fixed. In the case of DLBM and BDLBM, the shortage of tokens in the first bucket results in a large number of marked cells competing for free tokens in the second bucket, thus explaining the higher loss of tagged cells. In the case of BDLBP and BDLBP-DT, the shortage of tokens in the first token pool increases the chances of untagged



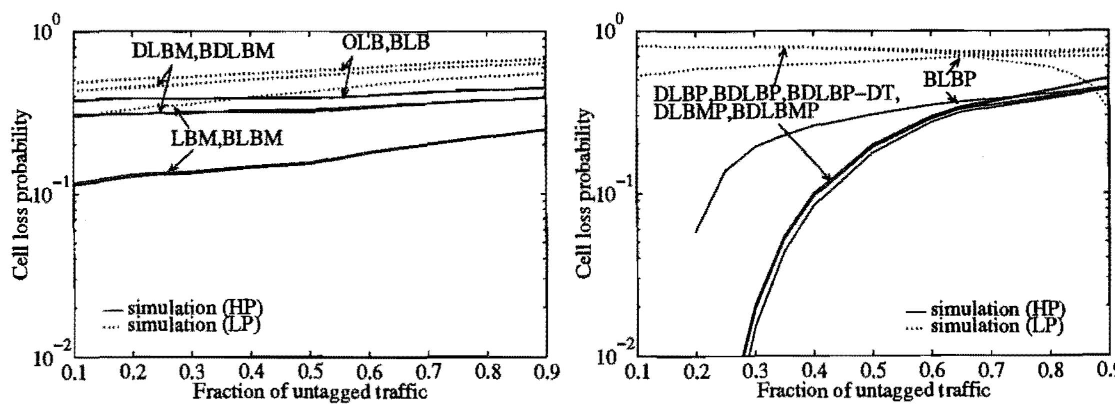
(a) Non-priority schemes (b) Priority schemes

Figure 4.28 Cell loss probability at the UPC versus untagged traffic ratio for $B = 3416$ and $f_R = 1.0$.



(a) Non-priority schemes (b) Priority schemes

Figure 4.29 Cell loss probability at the multiplexer versus untagged traffic ratio for $B = 3416$ and $f_R = 1.0$.



(a) Non-priority schemes (b) Priority schemes

Figure 4.30 End-to-end cell loss probability versus untagged traffic ratio for $B = 3416$ and $f_R = 1.0$.

cells queueing in the input buffer, which, in turn, increases the chance of the input buffer being full and tagged cells being discarded.

(b) Cell Loss at the Multiplexer

Figure 4.29(a) illustrates the performance of the multiplexer fed by policed sources. In general, as η increases, the cell loss probability for high and low priority traffic increases. Comparing the performance of the multiplexer in the case of policed and unpoliced sources, one can see that the improvement obtained by policing sources in the case of non-priority schemes is nearly constant for all values of η. For the priority schemes, except BLB, only small improvement in the performance of multiplexer is observed at low values of η; see Figure 4.29(b). This is because $\eta < f_S$, hence the token arrival rate is much higher than the untagged cell arrival rate, causing the leaky bucket to appear transparent to the arriving cells. For large $\eta, \eta \geq f_S$, cells arrive faster than the tokens. This results in a large number of untagged cells being discarded at the leaky bucket, and leaves very small number of cells to be discarded by the mutliplexer.

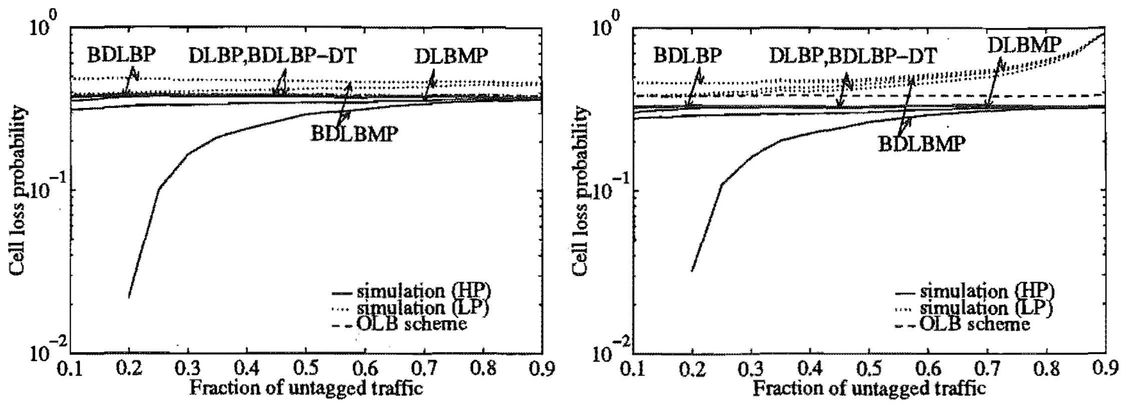
(c) End-to-End Cell Loss Probability

The results in Figure 4.30 illustrate the end-to-end performance for non-priority and priority schemes. The end-to-end performance under the priority schemes, see Figure 4.30(b), is very much the same as that at the UPC. On the other hand, the different quality of service offered to low and high priority cells in the case of OLB, BLB, DLBM, and BDLBM, see Figure 4.30(a), is due to the partial buffer sharing used at the multiplexer. For LBM and BLBM, marked traffic increases the overall low priority traffic arriving at the multiplexer, which in turn increases the cell loss probability of low priority cells at the multiplexer and the cell loss probability of high priority cells on an end-to-end basis. For example, simulation results for LBM show that the percentage of discarded low priority cells at the multiplexer coming from the marked traffic is only 4% when $\eta = 0.1$, while equals 75% when $\eta = 0.9$. Since LBM and BLBM do not differentiate between tagged and untagged cells, this means that, on average, the discarded low priority cells coming from the untagged traffic will be half of the values quoted.

Variation of f_S/η Ratio

So far, the results in the previous section show the effect of varying the percentage of untagged traffic, while keeping f_S constant, on the cell loss probabilities for both high and low priority traffic. In general, the performance of untagged traffic in priority schemes are expected to be better than that in non-priority schemes when $f_S > \eta$, and vice versa. In order to provide a clearer relationship, Figure 4.31 shows the cell loss probability as a function of η, while keeping the ratio f_S/η constant. The results are compared with those under OLB.

In Figure 4.31(a), we see that the loss probabilities for tagged and untagged traffic are invariant against the variation of η so long as the ratio f_S/η is kept constant at 1.0. The performance comparison between OLB and the priority schemes, except BDLBM, shows little improvement of the untagged traffic for $f_S/\eta = 1.0$, since at this ratio level, the token rates and the bucket depths for the first and the second token pools are chosen proportionate to the η value. By reserving 10% more tokens for the untagged cells through setting $f_S/\eta = 1.1$, see Figure 4.31(b), the priority



(a) $f_S/\eta = 1.0$ (b) $f_S/\eta = 1.1$

Figure 4.31 Cell loss probability at the UPC versus the ratio f_S/η for $B = 3416$ and $f_R = 1.0$.

schemes offer much better performance than OLB. In this case, the performance of untagged traffic is still invariant with respect to η, while cell loss probability of tagged traffic increases with η. The high cell loss probability for tagged traffic for large η is due to the diminishing of the token rate and bucket depth chosen for the second token pool.

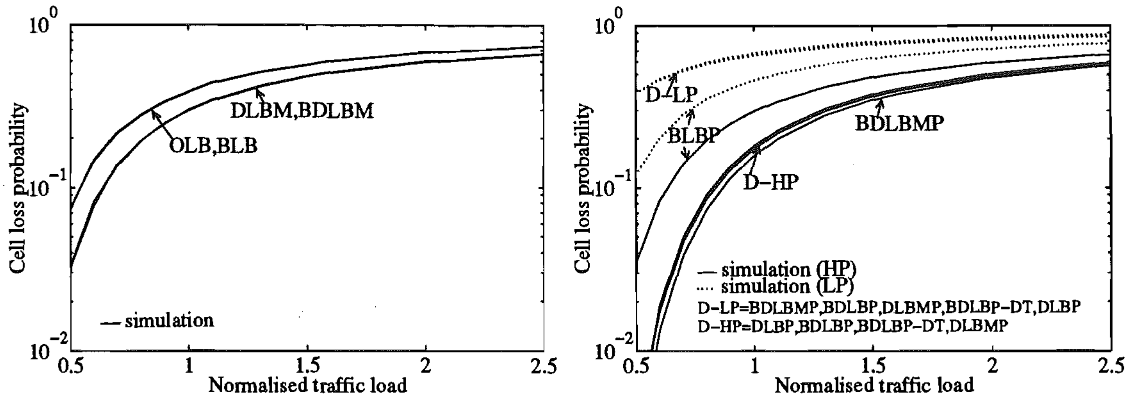
As shown in both figures, BDLBM offers much better improvement at small η because of the high rate at which tagged cells are discarded at the input buffer, hence allowing more opportunity for untagged cells to access tokens either from the first bucket or the second bucket.

Variation of Traffic Load

In this study, we vary the normalised traffic load f_L, that is, the ratio of the actual mean source rate R_m to the negotiated mean traffic rate R_m^*. This is achieved by increasing (decreasing) α, while keeping β constant, in order to increase (decrease) $R_m = \lambda\ \alpha/(\alpha + \beta)$, where α and β are the parameters of an IPP source; see Section 3.2.1.

(a) Cell Loss at the Leaky Bucket

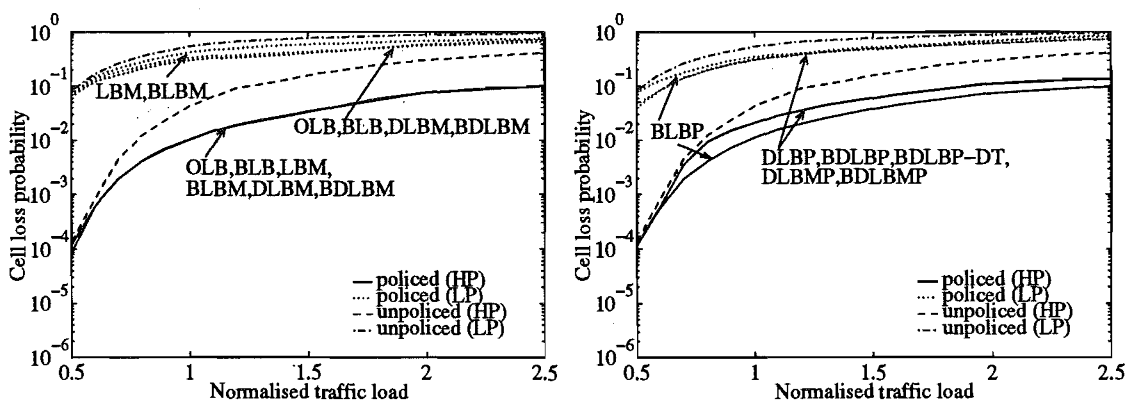
Figure 4.32(a) shows the cell loss probability at the UPC for non-priority schemes. Ideally, the cell loss probability should be zero when $R_m \leq R_m^*$ and approximately $(R_m - \gamma)/\gamma$ when $R_m > R_m^*$, where the token rate γ is chosen as $f_R R_m^*$ and $f_R \geq 1.0$, to allow some safety margin for minimising Error I, discussed in Section 4.4.1. The performance of the leaky bucket schemes shown in this figure is far from ideal. The high cell losses, observed when the normalised load is less than 1.0, indicate that average rate policing using the considered schemes is not effective as it will result in a high policing error, since the leaky buckets would discard cells that do not violate the negotiated mean source rate. For priority schemes, see Figure 4.32(b), the ideal behaviour is observed for the untagged traffic, where the loss probability is very low for $\eta R_m \leq \eta R_m^*$ and increases drastically for $\eta R_m > 0.5$ as $\eta R_m^* = 0.5$. Such ideal behaviour has been obtained due to the choice of $f_S > \eta$ and it indicates that priority schemes are better policing schemes for high priority traffic than non-priority schemes.



(a) Non-priority schemes

(b) Priority schemes

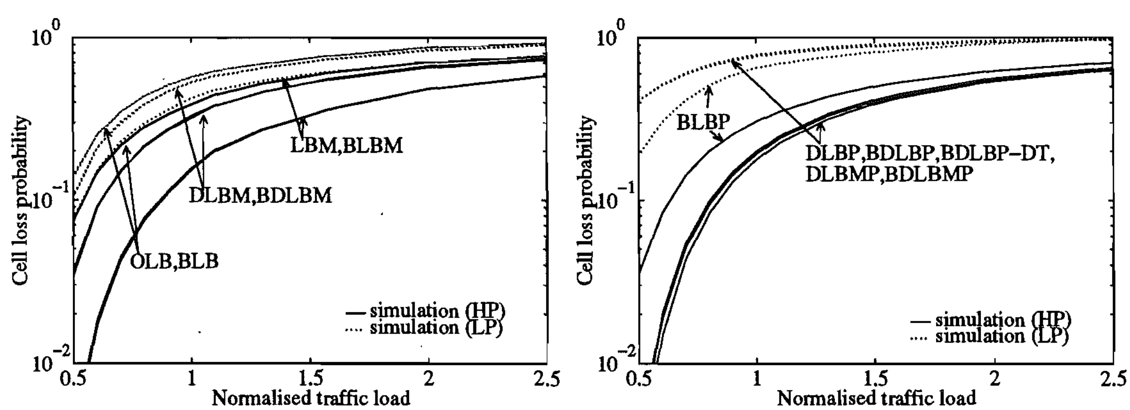
Figure 4.32 Cell loss probability at the UPC versus normalised traffic load for $B = 3416, f_R = 1.0$.



(a) Non-priority schemes

(b) Priority schemes

Figure 4.33 Cell loss probability at the multiplexer versus normalised traffic load for $B = 3416, f_R = 1.0$.



(a) Non-priority schemes

(b) Priority schemes

Figure 4.34 End-to-end cell loss probability versus normalised traffic load for $B = 3416, f_R = 1.0$.

(b) Cell Loss at the Multiplexer

Comparing the performance of a multiplexer fed by policed and unpoliced sources, (see the results presented in Figures 4.33(a) and (b)), one can see that the presence of a UPC allows the network to sustain a higher load for a specified cell loss probability. As seen from the graph, the cell loss probability for the policed sources flattens out for $R_m > 1.0$.

(c) End-to-End Cell Loss Probability

Observing the end-to-end performance curves for non-priority schemes in Figure 4.34(a), we can see that LBM and BLBM demonstrate the best performance. Unlike in other schemes, the softer policing action used by marking scheme prevent cells from being discarded wrongly at the leaky bucket. The performance for the priority schemes, see Figure 4.34(b), on the other hand, is very much the same as that observed at the UPC.

4.7.3 Influence of Correlation Between Priority Classes

Throughout this Chapter, we have assumed independent arrivals of high and low priority cells. However, in actual traffic from a codec, the cells tend to be generated periodically due to the need for gathering enough information to fill up a cell and furthermore a time correlation exists between the high and low priority cells. For example, in the voice coding process described in [Sriram *et al.*, 1991], a voice signal is sampled at a rate of 32 kbps with each sample being represented by 4 bits. The two most significant bits are packed into a cell marked with high priority and the other two are packed into a cell marked with low priority as shown in Figure 4.35. In order to fill a cell with 44 byte payload, the encoder is required to collect 22 ms sample. This means that the cell pair are generated every 22 ms and the high and low priority cells are generated in pairs and hence they are correlated.

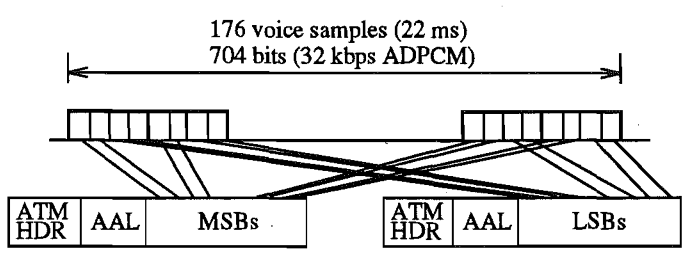


Figure 4.35 Voice coding method.

A correlation between the high and low priority cell arrivals also exists due to the tagging of excess cells by a user terminal implementing a marking scheme. It would be reasonable to assume that the instantaneous tagging probability is state dependent (i.e. the more bursty the traffic the larger the number of tagged cells). The effects of such state-dependency can be studied using a user policer (implementing an LBM scheme) at the user terminals to police the generated cells which only have a single priority class. Then, the UPC at the access node will encounter the arriving traffic comprising untagged and tagged cells.

For our study, the token rate of the user policer is taken to be equal to the mean source rate, while its bucket depth is chosen to be 2000. Using (4.10), it is found that this would make the long term fraction of untagged traffic to the total traffic approximately equal to 0.5. Figure 4.36 shows the cell loss probability at the UPC for both varying bucket depth and varying split factor.

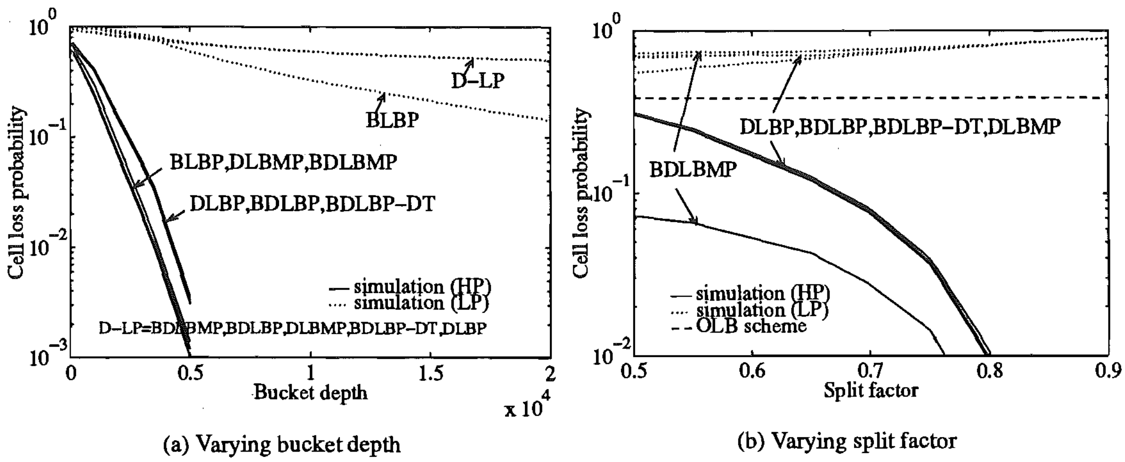


Figure 4.36 Cell loss probability at the UPC for priority schemes when $f_R = 1.0$.

Comparing Figure 4.36(a) with Figure 4.20(b), one can see that the cell loss probabilities obtained when considering the correlation between high and low priority traffic are lower than those obtained when assuming independent arrivals of high and low priority traffic. The same conclusion can also be drawn when comparing Figure 4.36(b) with Figure 4.26(a). The lower probabilities in the former case are due to the limitation imposed by the user policer on the maximum number of untagged cells that can arrive consecutively as a burst. Comparing the figures, we also notice that the order of performance offered by the leaky buckets are the same in both cases; BDLBMP offers the best performance while DLBP offers the worst performance among the priority schemes of dual bucket class.

Two reasonable conclusions may be drawn from the results presented in this section. Firstly, the cell loss probabilities obtained in previous sections under the independent arrival assumption are upper bounds on the true loss probabilities. Secondly, the performance ranking of the various leaky buckets is also unchanged by this assumption.

4.8 Conclusion

In this Chapter, we have proposed four modifications to existing leaky bucket schemes for policing mixed connections: DLBP, BDLBP, BDLBP-DT, and BDLBMP. These new schemes have been discussed and analysed along with previously known ones. Performance of all schemes was studied analytically, applying the approach presented in Chapter 3. Simulation results were used to verify the analytical solutions and generally they showed good agreements. The approximations used in analysing some of the schemes, such as BDLBM and BDLBP, were also shown to yield a good accuracy.

Comparative studies of the performance offered by the schemes were carried out by means of simulation techniques. Loss probabilities for untagged and tagged traffic were used as performance measures for the leaky buckets in policing an average rate of video sources generating pretagged traffic. The simulation results confirmed the difficulties OLB has in policing the average rate. To achieve a low discard rate, OLB requires very large safety margins in terms of having the token rate and the bucket depth to be considerably larger than the mean traffic rate and the mean burst length of the source. We showed that the benefit of having an additional buffer as in BLB for reducing these safety margins is negligible when dealing with bursty sources. The marking schemes, on the other hand, offer substantial improvements. Indeed, LBM and BLBM appeared to offer the best performance among non-priority schemes.

The different treatment given to untagged and tagged cells in priority schemes offers even better improvement. In the BLBP scheme preferential service can be given to untagged traffic by limiting the access of tagged traffic, hence limiting their competition for the input buffer space and tokens. This results in much greater improvement of non-priority BLB by BLBP. On the other hand, the additional token pool in dual bucket schemes, e.g. in BDLBP, offers an extra degree of freedom for controlling the rate of untagged and tagged trafffic separately. For the same total token rate γ and bucket depth B as those in OLB, a larger portion (measured by the *split factor* f_S) of the token rate and bucket depth can be given to the first bucket used for policing untagged traffic than to the second bucket. Results have shown that by having $f_S > \eta$, the priority scheme allows a typical quality of service requirement for untagged traffic, say 10^{-9}, to be achieved with a lower normalised token rate than that required in OLB. In general, we showed that the priority schemes perform better than the non-priority ones.

The similar level of performance offered by priority schemes with and without marking implies that cell marking at the leaky buckets is not essential for improving the services offered to high priority cells. For this reason, and because the network does not know the significance of cells when they are marked, we claim that any marking scheme should not be implemented in the network and that process should be relegated to the users. The network should only police the traffic from users by using non-marking schemes, such as the BLBP, DLBP, BDLBP, as part of the *user-network policing* scheme, proposed in [Hartanto and Sirisena, 1993a], while users can police their traffic appropriately and *selectively mark* any excess cells. Such an implementation is obviously favourable from the users' point of view as it offers them more control over their traffic. Using this solution, we can also maximise the network utilisation by properly selecting the parameters of the second bucket, as discussed in [Hartanto and Sirisena, 1993a], or by not policing the tagged traffic at all. These approaches allow the tagged cells to make use of any idle bandwidth within the network, which was the original intention of marking excess cells by the network. However, either approach does not exclude the possibility of tagged cells overloading the network and hence degrading not only performance offered to mixed connections but also to pure ones. In order to overcome this drawback, there is a need for establishing a buffer management scheme, which could protect the high priority cells against the overload created by tagged traffic, to be located at the convergence points of the traffic. This issue is discussed in the next Chapter.

Chapter 5

PROTECTIVE OUTPUT BUFFER POLICIES AT ATM SWITCHES

In Chapter 3, we have classified connections as: *pure* and *mixed.* In a mixed connection, users can pretag as low loss priority cells containing less essential coded information or cells violating the negotiated traffic parameters. This allows users to reduce the amount of bandwidth required by a connection and enables the network to admit more connections.

When the cells from both classes of connections converge at an ATM switch, two types of buffer policies can be used to resolve the competition for network resources, namely *separate buffer* and *shared buffer* policies. In the former type, separate buffers are reserved for cells from each class of connection and a buffer space unused by one class can not be used by the other. In the latter type, cells from pure connections are treated in the same way as high priority cells from mixed connections, and the high priority cells are queued in the same buffer as the low priority cells, although they are treated differently in terms of the access rights to the buffer.

The required buffer sizes are chosen to meet the QoS of each priority class, given the expected maximum load. However, if the network does not police the low priority traffic generated by the users, for example by allowing the cells to bypass the second bucket in DLBP or BDLBP (see Chapter 4), the buffer capacity may be exceeded causing an overload. Such an overload will not only degrade the QoS of high priority traffic in mixed connections, but also the QoS of traffic coming from pure connections, depending on the buffer policy used. We refer to this phenomenon as *priority interference effect.*

The degradation of QoS offered for the traffic of pure connections is obviously *unfair* for these connections, since they are strictly controlled either at the user end or at the network's access node to guarantee the cell delivery. This implies a need for buffering policies which can guarantee the QoS of traffic coming from pure connections irrespectively of the traffic intensity and arrival patterns of low priority cells, while preserving cell sequences. Such policies are termed *protective policies* by Cidon *et al.* [1993]. It is of our interest in this Chapter to investigate a number of classical as well as recently proposed buffer management policies with regard to their abilities in protecting traffic of pure connections against the overload of low priority traffic of mixed connections, while admitting as many low priority cells as possible when the offered traffic of pure connections is low.

Here, we consider both separate and shared buffer policies. Under a separate buffer policy, the protection for traffic of pure connections may not necessarily imply the protection for high priority traffic of mixed connections. Since the additional low priority traffic, which causes the overload, comes only from mixed connections, it is conceivable to allow the overload to have some limited effects on the QoS of high priority traffic from mixed connections, in order to discourage users from using cell marking excessively. Under a shared buffer policy, the traffic coming from pure connections is treated in the same way as the high priority traffic of mixed connections.

We start this Chapter with a survey of recent studies in the area of buffer management schemes employing loss priority, see Section 5.1. The model of an ATM switch indicating the location of the buffer management systems is described in Section 5.2, followed by a proposed definition of protection criterion. The criterion is more general than that defined in [Cidon *et al.*, 1993], to allow comparisons among various buffer management schemes which use not only shared policies but also separate policies. The details of four classical buffer management schemes, namely *complete buffer sharing (CBS)*, *partial buffer sharing (PBS)*, *push-out (PO)*, and *route separation (RS)*, are presented in Section 5.3. The first three schemes use a shared buffer policy, while the last one implements a separate buffer policy. Discrete-time analyses of the schemes are used to compare their admissible levels of low priority traffic and their protection against the overload caused by low priority traffic.

Investigations of some recently proposed buffer management schemes for improving the level of admissible low priority traffic are presented in Sections 5.4 and 5.5. In Section 5.4, we propose a modification to RS, called a *dual queues with cyclic service (DQCS)*, to allow some sharing of unused bandwidth between the different connection classes. The protection provided by this buffer policy is studied analytically in discrete time. In Section 5.5, we study the level of protection of two recent improvements to the PBS and PO schemes, and two recent proposal of protective buffer policies. Comparison of the schemes is carried out using simulation. Section 5.6 concludes this Chapter.

5.1 Review of Related Work

In the study of multi-class connections, loss priorities have played an important role in distinguishing cells from connections with different loss requirements within the network. Different loss priorities can be implemented either implicitly or explicitly.

The implicit introduction of loss priorities can be facilitated by using virtual paths, i.e. separate virtual paths can be set up for carrying connections having different loss requirements [Anido, 1989], where all cells within each connection have the same loss priority level. The boundary between the virtual paths can be fixed or movable. With a fixed boundary, no sharing of bandwidth is allowed, whereas with a movable boundary, the boundary can move depending on the load conditions in the virtual paths.

On the other hand, the explicit alternative is facilitated by identifying cell loss priorities within the ATM cell header. This identification allows cells of the same connection to have different loss priorities. The implementation of explicit cell loss priorities can be done through a number of buffer management schemes, such as nested threshold and push-out schemes.

In a *nested threshold* scheme, a common buffer of size K is provided for all cells of different priority classes and the buffer sharing is controlled by a set of discarding thresholds. Let K_i denote the discarding threshold for class i traffic and assume class i has priority over class $i-1$, i.e. cells from class i are less likely to be discarded than those belonging to class $i - 1$. An arriving cell of class i will only be admitted to the buffer if less than K_i cells have already been queued, otherwise it will be discarded. The class with the highest priority is given access to the entire buffer. A method for determining the optimal sets of discarding thresholds for the schemes was presented in [Petr and Frost, 1991].

Special cases of nested threshold schemes for two priority classes (low and high priority) include *partial buffer sharing (PBS)* [Kroner, 1990] and *limited low priority cell* policy [Bala *et al.*, 1990]. In the partial buffer sharing policy, low priority cells are accepted only if the total number of cells in the buffer is less than K_l, where $K_l < K$ (K is the buffer size), whereas in the limited low priority policy, low priority cells are accepted if the number of low priority cells in the buffer is less than K_l, irrespective of the number of high priority cells in the queue. Bala *et al.* [1990] compared the performance of the two policies and concluded that the former policy is preferred as it is less sensitive to a change of the buffer threshold K_l and hence is more robust.

The PBS scheme has been studied fairly extensively by assuming different traffic models and by using various analytical techniques. Kroner [1990] examined the bandwidth saving from the use of loss priorities when assuming independent Poisson arrival streams for both high and low priority cells. An ON/OFF model was assumed by Bonomi *et al.* [1990], with low priority cells arriving according to a Bernoulli trial process. Similar models were used by Le Boudec [1991] and Elwalid and Mitra [1992], who studied the queues based on matrix geometric and fluid flow approximation, respectively. A more general Markovian point processes were used by Garcia and Casals [1992], who analysed the queues using fluid flow approximation.

The *push-out (PO)* or *overwriting* scheme, on the other hand, was proposed independently by Sumita and Ozawa [1988] and Hebuterne and Gravey [1989]. In the scheme, low priority cells have the same access rights to the entire buffer as high priority cells. However, if an arriving high priority cell finds the buffer full, then one of the low priority cells which are already in the buffer will be replaced by the arriving cell. The replacement strategies, such as FIFO (first-in-first-out) and LIFO (last-in-first-out), specify which low priority cell is pushed out to make room for the newly arriving high priority cell. The strategy can also be generalised to allow a low priority cell arriving to the full buffer to replace another low priority cell already in the buffer. Kroner [1990] provided some surveys of existing studies based on different replacement strategies; Hebuterne and Gravey [1989] studied the push-out scheme with LIFO replacement strategy assuming Poisson arrival processes; Saito *et al.* [1991] studied the scheme assuming an MMPP input process; while Czachorski *et al.* [1992] investigated the scheme by applying a diffusion model.

Improvements to the PBS and PO schemes, with the aim of improving the performance of lower priority traffic, have also been proposed in recent years. Yau and Pawlikowski [1992] proposed a nested threshold with suspended execution scheme to delay the discarding of an arriving cell of class i when the queue length is larger than a buffer threshold K_i, but the buffer is not full. Beraldi and Maranao [1992] proposed the *limited push-out (LPO)* mechanism where the buffer is divided into two parts, with push-out policy being applied only in one part and no-priority policy being applied in the other part. Tipper *et al.* [1993] proposed two combinations of push-out and nested threshold mechanisms for controlling the QoS of high and low priority traffic separately, and analysed the schemes assuming a Bernoulli input model. The one called *threshold push-out (TPO)* scheme has been shown to offer the best improvement to the push-out scheme.

Most of the studies listed so far were concerned with determining acceptable load regions as a function of the cell loss requirements of each priority class. The issue of protecting high priority traffic against overload from low priority traffic was considered recently by Cidon *et al.* [1993], who identified and evaluated various classical shared buffer policies, such as push-out, partial buffer sharing and limited low priority, and concluded that the partial buffer sharing is the only protective buffer policy. Several new policies, which improve the performance of low priority traffic while remaining protective, were also proposed. Among these policies, the *simulated protective policy (SPP)* and the *extended simulated protective policy (ESPP)* were shown to offer the largest improvement of service quality for low priorty traffic. Anggawijaya [1994] modified the ESPP scheme to delay the discarding of low priority cells until it is certain that these cells can not be served. The modified scheme is referred to as ESPP with suspended execution (ESPP/SE).

The application of the protection criterion defined in [Cidon *et al.*, 1993] for other recently proposed shared schemes, such as the LPO and TPO schemes, does not appear to have been considered previously. Such analysis may not be as straightforward as for the PBS and PO policies, because the new schemes are combinations of both policies. Moreover, the protection criterion, which was defined for a shared buffer policy, and also the quantitative comparison between the levels of protection offered by different policies, which was based on a fixed buffer size [Cidon *et al.*, 1993], may not be applicable for separate buffer policies, because such policies normally require larger buffer sizes in order to satisfy a given cell loss probability. This indicates a need for more general criterion for evaluating the level of protection of a buffer policy and for making comparisons with other buffer policies.

A more general approach for comparing the levels of protection between shared buffer and separate buffer policies was used in [Hartanto *et al.*, 1991]. In the paper, buffer thresholds were optimised to meet the quality of services of high and low priority traffic for a given maximum traffic load. The offered load coming from low priority traffic was increased and the resulting cell loss probabilities of high priority traffic were compared. A scheme was said to offer protection if it could limit the increase in the cell loss probability of high priority traffic. We will use the same approach in this Chapter and will define a more general protection criterion in Section 5.2.2.

5.2 ATM Switch Model

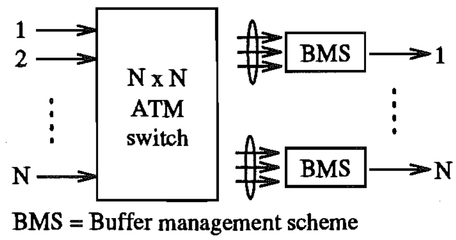


Figure 5.1 ATM switch with output buffer.

We consider an $N \times N$ ATM switch as shown in Figure 5.1. Each output of the switch is supported by a buffer management system which comprises one or multiple buffers, an enqueuer to control the access of the buffers, and a scheduler or a server to control the cell transmission to the output link. As an example, a functional block for a buffer management scheme with two buffers is shown in Figure 5.2. The enqueuer identifies the cells according to their loss priority classes and queues them in their respective buffers according to some buffering policies, while the scheduler visits each queue according to a given service scheduling policy.

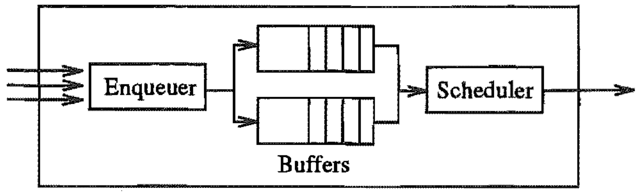


Figure 5.2 Functional schematic of a buffer management system.

In the following sections, we will describe the traffic models used for evaluating the protection levels of a buffer policy and define a protection criterion used when interpreting the results.

5.2.1 Traffic Models

For studying the performance of the buffer management schemes, we assume that the time in the system is slotted, and the length of slot is equal to the transmission time of ATM cell. Let N_p and N_m be the number of pure and mixed connections, respectively, where $N_p + N_m = N$. Cells from pure and mixed connections are assumed to arrive at their respective inputs according to stationary Bernoulli processes with traffic load ρ_p and ρ_m, respectively. We assume that the connections of the same class are homogeneous. This means the offered load of each connection is simply $\rho_x/N_x, x = p, m$. The high and low priority cells within a mixed connection are assumed to arrive independently, where the ratio of high priority traffic to total traffic is η_m.

Let $\delta_p(k,i,j)$ and $\delta_m(k,i,j)$ denote the existence of a cell of pure and mixed connections, respectively, at an input i destined to an output j in slot k. Each of them can assume value 0 if no cell exists and 1 if cell exists. If we assume that the switch is operating at N times higher speed than the input trunks, then the number of cells from pure or mixed connections arriving at an output j in slot k is equal to $A_x(k,j) = \sum_{i=1}^{N_x} \delta_x(k,i,j)$, where $x = p, m$. The total number of cells arriving at the output is simply $A(k,j) = A_p(k,j) + A_m(k,j)$. Based on the assumption that each cell is equally likely to be destined to any of N outputs, we can drop the dependence on j and hence $A_x(k,j) = A_x(k)$.

The probability mass function $\{a_x(i)\}$ of $A_x(k)$ is given as

$$a_x(i) = Pr[A_x(k) = i] = \begin{cases} \begin{pmatrix} N_x \\ i \end{pmatrix} \left(\frac{\rho_p}{N_x}\right)^i \left(1 - \frac{\rho_x}{N_x}\right)^{N_x - i} & \text{for } 0 \le i \le N_x \\ 0 & \text{otherwise} \end{cases} \quad (5.1)$$

Due to the assumed independence among the connections, the probability mass function $\{a(i)\}$ of $A(k)$ can be written as a discrete convolution of $\{a_p(i)\}$ and $\{a_m(i)\}$, i.e.

$$a(i) = a_p(i) * a_m(i) \quad (5.2)$$

5.2.2 Protection Criterion

Prior to defining a protection criterion, we will first consider the following procedure for dimensioning the buffer requirements at the ATM switch for given maximum load and quality of service requirements [Hartanto *et al.*, 1991].

Assume that the switch is dimensioned for a maximum load of ρ_{max}. This load comprises $\rho_{p(max)}$ (load generated by pure connections), $\rho_{mh(max)}$ (high priority load generated by mixed connections) and $\rho_{ml(max)}$ (low priority load generated by mixed connections). We define an *overload* by a given traffic of type $x(x = p, mh, ml)$ if $\rho_x > \rho_{x(max)}$. Let the QoS requirements for traffic of pure connections, high priority traffic and low priority traffic of mixed connections be QoS_p, QoS_{mh}, and QoS_{ml}, respectively. For a shared buffer policy, the offered load for high priority traffic will be $\rho_{p(max)} + \rho_{mh(max)}$ and $QoS_p = QoS_{mh}$, while for a separate buffer policy, the offered load for traffic of pure connections and high priority traffic of mixed connections are individually considered and their QoS requirements may or may not be equal. We dimension the buffer size and choose the discarding thresholds for a given buffer management scheme in order to meet all QoS requirements at the given maximum operating load. For some schemes, such as the PO scheme, where QoS of high and low priority traffic can not be satisfied at the same time, we will dimension the buffer sizes to meet the most stringent QoS requirements.

Assume that the buffer size of a scheme is fixed and that any additional buffer threshold is adaptive. When the high priority load is below its maximum value, it is desirable that we can accept as many low priority cells as possible, without the need to adjust the buffer thresholds while still satisfying the high priority QoS. On the other hand, when the high priority load is at maximum, we should be able to limit the degradation of high priority QoS against any overload by low priority traffic, i.e when $\rho_{ml} > \rho_{ml(max)}$. Based on these observations, we define our protection criterion as follows:

> A buffer policy is *protective* if it can satisfy the QoS of high priority traffic against any level of overload from low priority traffic, when the high priority traffic load is less than or equal to its maximum load, i.e. $\rho_p \leq \rho_{p(max)}$ and $\rho_{mh} \leq \rho_{mh(max)}$. A policy is also *protective* if it allows *adjustment of its buffer threshold*, without increasing its buffer size, to maintain the QoS of high priority traffic when the high priority load is at maximum.

A buffer policy can be protective by dedicating resources exclusively to high priority traffic, but if any idle resources can not be utilised by low priority traffic, then we get poor utilisation of network resources. On the other hand, a buffer policy may admit a large number of low priority traffic when the offered load is below the maximum load, and by doing this it can weaken protection of the high priority traffic when overload does occur. This implies that there is a tradeoff between the admissible level of low priority traffic and the level of protection of high priority traffic. Considering such a tradeoff, we define the *best buffer policy* to be the one which stays protective against the overload by low priority traffic when the high priority traffic is at its maximum, while admitting the most low priority traffic when the high priority traffic load is less than maximum. Based on these definitions, in the following sections we will compare various existing buffer management schemes and determine their levels of protection.

5.3 Classical Buffer Management Schemes

In this section we describe and analyse four classical buffer management schemes. Recently proposed improvements to the schemes will be presented in Sections 5.4 and 5.5.

5.3.1 Complete Buffer Sharing Scheme

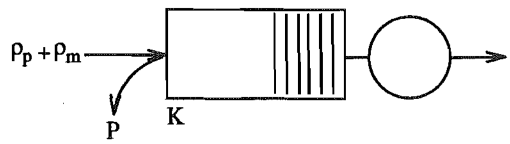


Figure 5.3 Complete buffer sharing (CBS) scheme.

In a *complete buffer sharing (CBS)* scheme depicted in Figure 5.3, there is no differentiation between high and low priority cells. All cells are queued in the same buffer of size K, served in a first-in first-out (FIFO) order and no delay priority is distinguished. We evaluate this scheme as a reference model for the remaining schemes considered here.

We follow a discrete time analysis similar to that in Section 3.4. The following assumptions are made:

(A5.1) Cell departure takes place at the beginning of a slot.

(A5.2) New cells are admitted into the buffer in batches at the end of each slot.

(A5.3) The system is in statistical equilibrium.

Based on these assumptions, the relationship between the states of the queue immediately prior to and immediately after the beginning of the kth slot, denoted by $Q^-(k)$ and $Q^+(k)$, respectively, is as follows

$$Q^+(k) = \max(Q^-(k) - 1, 0) \tag{5.3}$$

$$Q^-(k+1) = \min(Q^+(k) + A(k), K) \tag{5.4}$$

Let $\{q^-(k,j)\}$ and $\{q^+(k,j)\}, j = 0,1,2,\ldots,K$, be the probability mass functions for $Q^-(k)$ and $Q^+(k)$, respectively, and $\{a(i)\}, i = 0,1,2,\ldots,N$, be the probability mass function of arrival probabilities $A(k)$. Using the operators Σ and $*$ defined in (3.31)-(3.33), we have

$$\begin{array}{lcll} q^+(k,j) & = & \Sigma_0(q^-(k,j+1)) & \text{if } 0 \le j \le K-1 \\ q^-(k+1,j) & = & \Sigma^K(q^+(k,j) * a(j)) & \text{if } 0 \le j \le K \end{array} \tag{5.5}$$

Using the probability mass function $\{a(j)\}$ given by (5.2), and starting from an empty and idle system, i.e. assuming that

$$q^+(0,j) = \begin{cases} 1 & \text{for } j = 0 \\ 0 & \text{otherwise} \end{cases} \tag{5.6}$$

we can solve the queueing system iteratively for $k = 1, 2, \ldots$, stopping calculations at $k = k_{max}$ when

$$\left| \frac{q^+(k,j) - q^+(k-1,j)}{q^+(k,j)} \right| < \epsilon \tag{5.7}$$

with $\epsilon = 1 \times 10^{-6}$, which can be assumed to be the steady state conditions, i.e. $\lim_{k\to\infty} q^+(k,j) = q^+(j)$.

Since the departure is assumed to take place before any new arrivals being accounted, this means that the system can hold up to K cells in the buffer instead of $K+1$ as in Section 3.4.3. Hence cell blocking will occur if there are more than $K - j$ cells arriving in a batch when the queue length is j. This results in an average cell loss after unconditioning the queue length j as

$$L = \sum_{j=0}^{K-1} q^+(j) \sum_{i=K-j+1}^{N} [i - (K-j)]\, a(i) \tag{5.8}$$

The cell loss probability can then be obtained by dividing L by the offered load ρ, i.e.

$$P_{(CBS)} = \frac{L}{\rho} \tag{5.9}$$

5.3.2 Partial Buffer Sharing Scheme

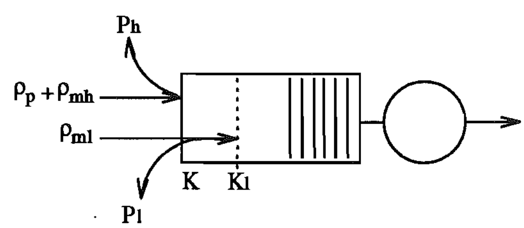


Figure 5.4 Partial buffer sharing (PBS) scheme.

The *partial buffer sharing (PBS)* scheme, depicted in Figure 5.4, has an additional buffer threshold K_l, which limits the access of low priority cells to a buffer of size K. Due to this limit, there is a need for distingushing the low priority cells from the high priority cells in mixed connections. The probability mass functions $\{a_{mh}(i)\}$ and $\{a_{ml}(i)\}$, $i = 0, 1, \ldots, N_m$, for high and low priority cells, respectively, can be derived from (5.1) as

$$a_{mh}(i) = \begin{cases} \sum_{m=i}^{N_m} a_m(i) \begin{pmatrix} m \\ i \end{pmatrix} \eta_m^i\, (1-\eta_m)^{m-i} & \text{if } 0 \le i \le N_m \\ 0 & \text{otherwise} \end{cases} \tag{5.10}$$

$$a_{ml}(i) = \begin{cases} \sum_{m=i}^{N_m} a(i) \begin{pmatrix} m \\ i \end{pmatrix} (1-\eta_m)^i\, \eta_m^{m-i} & \text{if } 0 \le i \le N_m \\ 0 & \text{otherwise} \end{cases} \tag{5.11}$$

We assume that cells from pure connections have the same priority level as high priority cells of mixed connections, so the total offered load of high priority cells is $\rho_h = \rho_p + \rho_{mh}$ and the ratio of high priority traffic to overall traffic load is equal to $\eta_h = \rho_h/\rho$. The overall probability mass functions $\{a_h(i)\}$ and $\{a_l(i)\}$ for high and low priority traffic, respectively, are simply

$$a_h(i) = a_p(i) * a_{mh}(i) \tag{5.12}$$

$$a_l(i) = a_{ml}(i) \tag{5.13}$$

Due to the access limitation on the low priority cells, the actual number of cells that are accepted into the buffer when $Q^-(k) \geq K_l$ comprises only the high priority cells, namely

$$a(i|j) = \begin{cases} a(i) & \text{if } 0 \leq j \leq K_l - 1 \\ a_h(i) & \text{if } K_l \leq j \leq K - 1 \end{cases} \tag{5.14}$$

which is the conditional probability that we have i cells arrive when j cells are in the queue.

Applying this condition when deriving the cell loss probability following the approach in the previous section, we find

$$P_{h(PBS)} = \sum_{j=0}^{K-1} q^+(j) \frac{L_h(j)}{\rho_h} \tag{5.15}$$

$$P_{l(PBS)} = \sum_{j=0}^{K-1} q^+(j) \frac{L_l(j)}{\rho_l} \tag{5.16}$$

where

$$L_h(j) = \begin{cases} \eta_h \sum_{i=K-j+1}^{N} [i - (K-j)]\, a(i) & \text{if } 0 \leq j \leq K_l - 1 \\ \sum_{i=K-j+1}^{N_m} [i - (K-j)]\, a_h(i) & \text{if } K_l \leq j \leq K - 1 \end{cases} \tag{5.17}$$

$$L_l(j) = \begin{cases} (1-\eta_h) \sum_{i=K-j+1}^{N} [i - (K-j)]\, a(i) & \text{if } 0 \leq j \leq K_l - 1 \\ \sum_{i=1}^{N_m} i\, a_l(i) & \text{if } K_l \leq j \leq K - 1 \end{cases} \tag{5.18}$$

5.3.3 Push-Out Scheme

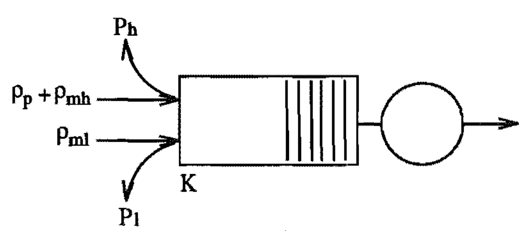


Figure 5.5 Push-out (PO) scheme.

In a *push-out (PO)* scheme, depicted in Figure 5.5, arriving low priority cells will have the same access rights as high priority cells. However, when k arriving high priority cells find no buffers available, then k low priority cells will be pushed out from the queue. If there are only $l(l < k)$ low priority cells in the buffer, then $k - l$ arriving high priority cells will be lost. The probability mass functions for both high and low priority cells are the same as for PBS, given in (5.10) and (5.11).

A common analytical approach, used in [Hebuterne and Gravey, 1989; Kroner, 1990; Kroner *et al.*, 1991; Saito *et al.*, 1991], is to find the loss probability of the low priority cells and then to use the conservation law given in [Sumita and Ozawa, 1988] for finding the loss probability for the high priority cells. This approach, however, requires the need to formulate the joint probability of the number of cells in the queueing system, and the remaining service time of the momentary served cell. Since such a formulation can be quite complex, we decided to derive the loss probabilities differently, following a procedure similar to that discussed in [Czachorski *et al.*, 1992].

The conservation law derived by Sumita and Ozawa [1988] states that the number of low priority cells pushed out from the buffer is equal to the number of high priority cells pushing them out, hence the total throughput γ of the system with and without push-out scheme should be the same. In the system without push-out scheme, the total throughput is equal to the sum of throughput of the low and high priority traffic, namely

$$\gamma = \gamma_h + \gamma_l \tag{5.19}$$

where $\gamma_h = \rho_h(1 - P_{(CBS)})$, $\gamma_l = \rho_l(1 - P_{(CBS)})$, and $P_{(CBS)}$ is the cell loss probability for the CBS scheme given in (5.9).

In the system with push-out scheme, replacements of low priority cells by high priority cells occur when the system is full. During such saturation period, the number of replacements depends on the number of low priority cells in the queue and the number of arriving high priority cells. We assume that any arriving low priority cells during that period of time will be discarded.

Since cell departure takes place before arrivals are accounted, arriving cells only see a maximum of $K - 1$ cells in the buffer. During the saturation period, the probability of having n_l low priority cells out of the cells in the queue is given by

$$p_l(n_l) = \binom{K-1}{n_l} \left(\frac{\gamma_h}{\gamma}\right)^{n_l} \left(1 - \frac{\gamma_h}{\gamma}\right)^{K-n_l-1} \tag{5.20}$$

For a batch arrival of $i(i \leq N)$ high priority cells, i replacements occur if there are at least i low priority cells within the queue, otherwise only n_l, $n_l < i$, replacements will take place if there are only n_l low priority cells in the queue. Based on these two factors, we can determine an average number of replacements as

$$L_{repl} = \sum_{n_l=1}^{K-1} p_l(n_l) \left[\sum_{i=0}^{n_l-1} i \, a_h(i) + n_l \sum_{i=n_l}^{N} a_h(i) \right] \tag{5.21}$$

Given the average number of high priority cells arriving per slot is ρ_h, the probability of high priority cells pushing out the low priority cells, denoted as ψ, can be obtained as the ratio of the average number of replacements to the average number of high priority cells arriving per slot, i.e.

$$\psi = L_{repl}/\rho_h \tag{5.22}$$

Due to these replacement, the throughput of high priority traffic will increase by $P_{(CBS)}\psi\rho_h$, while the throughput of low priority traffic will decrease by the same amount. Thus

$$\begin{aligned} \gamma_h &= (1 - P_{(CBS)})\rho_h + P_{(CBS)}\psi\rho_h \\ \gamma_l &= (1 - P_{(CBS)})\rho_l - P_{(CBS)}\psi\rho_h \end{aligned} \tag{5.23}$$

After the replacements, the conditional probability in (5.20) will change according to the new values of γ_h and γ_l, which in turns will change the value of ψ. This interdependency indicates the need for an iterative procedure to determine the final value of ψ, which then allows us to calculate the loss probabilities for both high and low priority cells, which are given as

$$\begin{aligned} P_{h(PO)} &= \frac{\rho_h - \gamma_h}{\rho_h} = (1-\psi)P_{(CBS)} \\ P_{l(PO)} &= \frac{\rho_l - \gamma_l}{\rho_l} = \left(1 + \frac{\rho_h}{\rho_l}\psi\right) P_{(CBS)} \end{aligned} \tag{5.24}$$

The iterative procedure can be formulated as follows.

Algorithm 5.1.

Step 1. Determine the loss probability $P_{(CBS)}$ for the system without replacements.

Step 2. Initialise the value of $\psi(0)$, i.e. $\psi(0) = 0.0$.

Step 3. Calculate the conditional probability $p_l(n_l)$ during the saturation period according to (5.20).

Step 4. Calculate the average number of replacements according to (5.21) and then calculate new value of $\psi(n)$ according to (5.22).

Step 5. Based on the new value of $\psi(n)$, estimate new values of $\gamma_h(n)$ and $\gamma_l(n)$ applying (5.23).

Step 6. Repeat steps 3-5 until $|(\psi(n) - \psi(n-1))/\psi(n)| \leq \epsilon$, where ϵ is the required precision (default value $\epsilon = 10^{-6}$).

Step 7. Calculate the loss probability from (5.24).

5.3.4 Route Separation Scheme

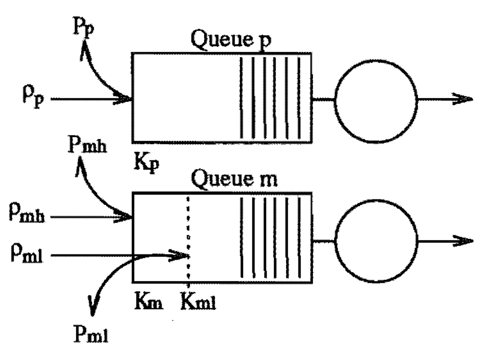


Figure 5.6 Route separation (RS) scheme.

The *route separation (RS)* scheme, depicted in Figure 5.6, is the simplest buffer management scheme for handling two connection classes, where cells from pure and mixed connections are queued in two separate buffers, a portion of link bandwidth is dedicated to each queue, and no sharing of bandwidth is allowed.

The cell loss probability for the queue of cells of pure connections (Queue p) can be evaluated as a CBS scheme with offered traffic being scaled by the proportion θ_p of output link allocated to the queue, namely $\rho_p = \rho_p/\theta_p$, whereas the queue of cells of mixed connections (Queue m) can be evaluated as in PBS with $\rho_m = \rho_m/\theta_m$. Thus

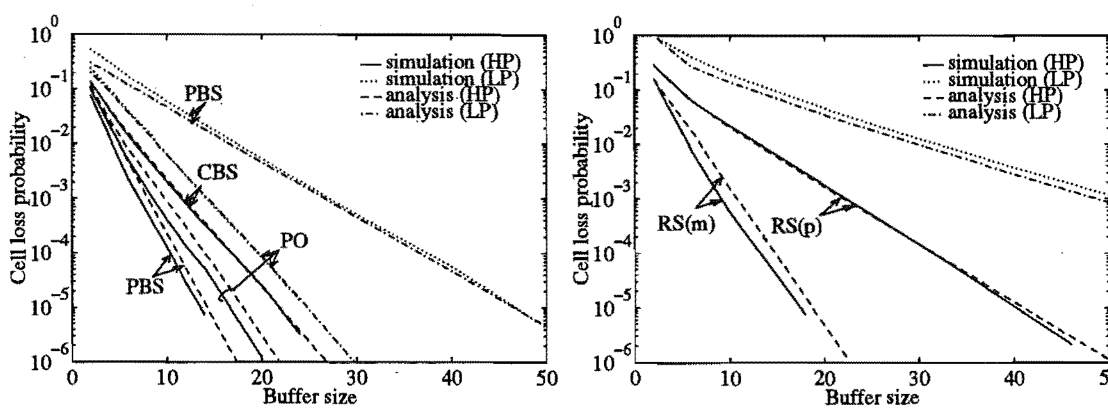
$$P_{p(RS)} = P_{(CBS)} \tag{5.25}$$

$$P_{mh(RS)} = P_{h(PBS)} \tag{5.26}$$

$$P_{ml(RS)} = P_{l(PBS)} \tag{5.27}$$

5.3.5 Analysis Verification

Before comparing the performance of the buffer management schemes, we will first verify analytical results against simulation results. All simulation results have 0.05 precision at 95% confidence intervals. As a reference we consider a switch size of 16×16. We assume that there are equal numbers of pure and mixed connections and that they offer equal amount of load. The ratio of high priority traffic to total mixed traffic is 0.5.



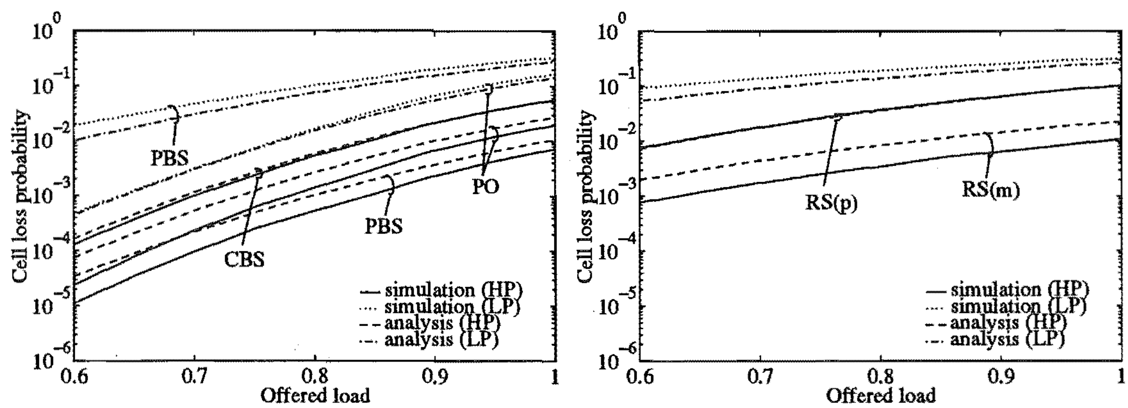
(a) Shared buffer policies

(b) Separate buffer policy

Figure 5.7 Cell loss probability for various buffer sizes with $\rho_p = 0.4$, $\rho_m = 0.4$ and $\eta_m = 0.5$.

In Figure 5.7 we compare the cell loss probabilities for varying buffer size with equal offered loads of 0.4 from pure and mixed connections, i.e. $\rho_p = \rho_m = 0.4$. From these results, we can see that the analytical results provide an excellent accuracy for CBS, and low priority traffic for PBS and PO, while they overestimate the loss probability for high priority traffic in PBS and PO. The accuracy of results for RS is similar to CBS for Queue p and to PBS for Queue m.

The discrepancy in the analytical results obtained for PBS is due to the fact that we do not take into account the transition between states $j < K_l$ to states $j \geq K_l$ as done in [Kroner, 1990]. This results in batch arrivals of low priority cells being accepted entirely to the buffer after the first cell has been accepted when $j \leq K_l$. For example, when the queue length is $K_l - 1$ and six low priority cells arrive, whereas only one cell should be accepted, in our analysis all six cells are accepted. This explains the discrepancy and also the fact that the analytically obtained loss probabilities for low priority traffic are lower than those obtained from simulation. On the other hand, the discrepancy in the results for PO, is due to the fact that the actual replacement strategy is not considered in the analytical model and is only approximated.



(a) Shared buffer policies

(b) Separate buffer policy

Figure 5.8 Cell loss probability for various offered loads with $K = 8$.

In Figure 5.8 we compare the cell loss probabilities for varying offered load with $K = 8$. Again, the results show a very good accuracy for low priority traffic, while they are less accurate for high priorty traffic. In general, the accuracy for high priority traffic tends to improve as the offered load increases.

5.3.6 Performance Comparisons

In this section, we compare the buffer management schemes in terms of the levels of admissible low priority traffic and the levels of protection of high priority traffic against overload by low priority traffic, for given buffer sizes.

We assume that the maximum operating load $\rho_{max} = 0.9$, with $\rho_{p(max)} = \rho_{mh(max)} = 0.4$, and $\rho_{ml(max)} = 0.1$. The QoS for pure connections (QoS_p) and for high priority cells of mixed connections (QoS_{mh}) are assumed to be the same and set at 10^{-9}, whereas the QoS for low priority cells of mixed connections (QoS_{ml}) is 10^{-3}. The buffer thresholds required are shown in Table 5.1.

	CBS	PO	PBS		RS
K	63	57	52	K_p	40
K_l	-	-	26	K_{mh}	65
				K_{ml}	43
Total	63	57	52		105

Table 5.1 Optimum buffer thresholds (QoS_p=QoS_{mh}=10^{-9}, QoS_{ml}=10^{-3}).

The table shows that among the shared buffer schemes, PBS requires the least buffer space due to its ability to satisfy exactly the QoS for both high and low priority traffic. For PO, we can either just satisfy the QoS for the high priority traffic or the low priority one. In the former case the cell loss probability for low priority would be much below QoS_{ml}. For example, in the analysed case the loss probability for low priority traffic reached 9.350×10^{-8}, which is four order of magnitude lower than necessary. As expected, RS scheme requires the largest buffer size, which is more than twice as many as under PBS scheme.

Admissible Levels of Low Priority Traffic

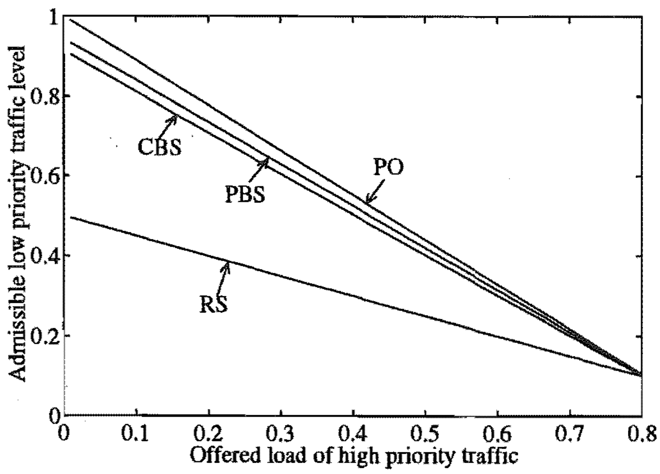
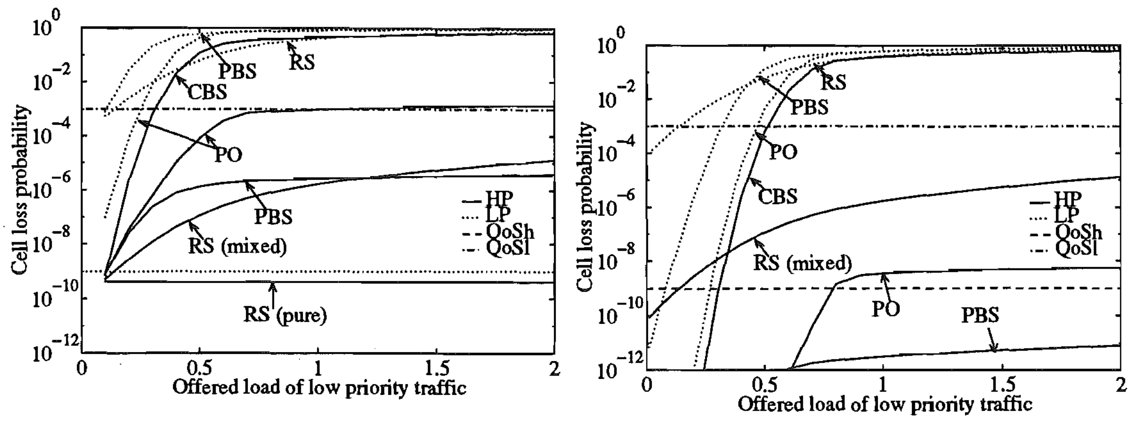


Figure 5.9 Admissible low priority traffic level for various high priority traffic load.

Figure 5.9 shows the admissible level of low priority or marked traffic as a function of high priority traffic load. As expected, RS scheme admits the least amount of low priority traffic due the fact that resources are not shared between classes of pure and mixed connections. Any idle bandwidth from the class of pure connections will be wasted. Other schemes, which allow low priority traffic to share bandwidth unused by the class of pure connections, admit more cells. The graph indicates that PO admits the largest amount of low priority traffic due to the fact that buffer spaces and transmission bandwidths are completely shared between high and low priority traffic when the buffer is not full. Partially excluding such sharing under PBS results in reduced level of admissible low priority traffic.

Protection Against Overload by Low Priority Traffic

Figure 5.10(a) shows the cell loss probabilities for all traffic classes as a function of the amount of overload of low priority traffic when the offered load of the traffic of pure connections and the high priority traffic of mixed connections is at their maximum. From the graphs, we can see that the pure traffic suffers from the overloading caused by low priority traffic in the case of buffer sharing schemes (CBS, PBS and PO). It degrades equally as the high priority traffic of mixed connections. It is obvious that CBS gives no protection against the overload due to the complete sharing of resources by all types of cells. Letting arriving high priority cells push out low priority cells from the buffer allows the PO scheme to offer better protection. The fact that CBS and PO schemes have no means to reduce the loss probability of high priority traffic below the required QoS implies that the schemes are non-protective.



(a) $\rho_p = 0.4, \rho_{mh} = 0.4$

(b) $\rho_p = 0.2, \rho_{mh} = 0.4$

Figure 5.10 Cell loss probability versus the offered load of low priority traffic.

PBS scheme offers the best protection among the shared buffer policies due to the reservation of $K - K_l$ buffer space for the high priority traffic at the expense of low priority traffic. The threshold K_l can be adjusted to allow the QoS of high priority class to be satisfied for a given low priority load ρ_{ml}. For example, when $\rho_{ml} = 2.0$, the QoS of high priority traffic will be satisfied if the threshold is adjusted to $K_l = 8$. Based on the protective criterion defined in Section 5.2.2, PBS buffer policy is protective.

As expected, under RS, the graph shows no degradation of performance for the traffic of pure connections due to non-sharing resources between the traffic classes. It is interesting to note that RS scheme offers better protection than PBS for the high priority traffic of mixed connections when $\rho_{ml} < 1.3$. This is due to larger buffer size used under RS. However, when the overload increases beyond 1.3, the loss performance of RS degrades faster than that of PBS. This is due to the fact that there is smaller buffer reservation for high priority traffic under RS than under PBS. Our results show that in order to reduce the loss probability of high priority traffic of mixed connections below the QoS_{mh} when $\rho_{ml} = 2.0$, RS can adjust K_{ml} to 22. Based on the protective criterion defined in Section 5.2.2, RS buffer policy is protective.

Figure 5.10(b) depicts the cell loss probabilities for all priority classes versus the amount of overload of low priority traffic when the traffic load of pure connections is below its maximum level. For the shared buffer policy, the total high priority offered load is 0.6. The graphs show that CBS and PO schemes are still unable to offer full protection for high priority traffic. The loss probability for high priority traffic under PO levels off about five orders of magnitude lower than that when $\rho_p + \rho_{mh} = 0.8$. However, it is still higher than the required QoS. This confirms that CBS and PO policies are non-protective.

The PBS scheme, on the other hand, offers better performance for both high and low priority traffic. The loss probability for high priority traffic stays below the required QoS, without the need for adjusting the buffer threshold. This again confirms that PBS policy is protective. RS scheme also offers much smaller loss probability for traffic of pure connections than the required QoS, while the loss probability for traffic of mixed connections is unchanged. Although this still means that RS policy is protective, it illustrates the drawback of the scheme in the way the protection is achieved since any idle bandwidth unused by traffic of pure connections can not be used by traffic of mixed connections.

Comparison Summary

In summary, we found that CBS and PO are non-protective, while PBS and RS are protective. This finding is consistent with that in [Cidon *et al.*, 1993]. Comparing PBS and RS, we have found PBS to be the best policy, because it admits more low priority cells and requires smaller buffer sizes. However, PBS requires adjustment of the buffer threshold in order to offer full protection for traffic of pure connections, when the high priority traffic load is at maximum. In this sense, RS offers better solution since it does not require such an adjustment. The desire to improve the level of low priority traffic admitted under RS leads us to propose the DQCS in the next section.

5.4 Improved Separate Buffer Policy

As concluded in the previous section, RS scheme provides the best protection for the traffic of pure connections but at the cost of admitting the fewest low priority cells, since any bandwidth unused by the class of pure connections can not be utilised by the class of mixed connections. In order to overcome this drawback, but yet to remain protective, we propose a modification to RS scheme, called *dual queues with cyclic service (DQCS)*. DQCS was initially proposed in [Hartanto *et al.*, 1991] for handling loss tolerable (e.g. voice) and loss sensitive (e.g. data) services. It allows the sharing of idle bandwidth between the connections, yet guaranteeing a minimum bandwidth allocation for each connection class. The scheme is further extended in [Hartanto and Sirisena, 1993a] to include an additional buffer threshold in the queue for loss tolerable services. We will use the same scheme for handling classes of pure and mixed connections, to allow the dynamic sharing of unused bandwidth between the classes. With bandwidth sharing, we expect that the queue will be depleted faster and hence that the cell loss probability will decrease.

5.4.1 Model Description

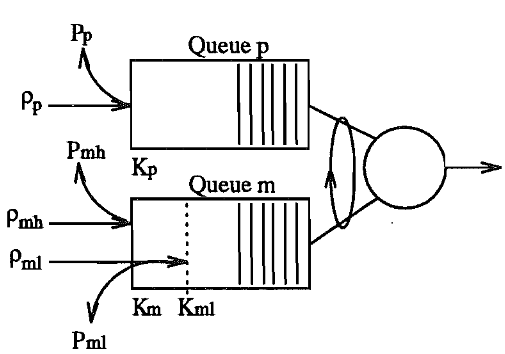


Figure 5.11 Dual queues with limited cyclic service (DQCS) scheme.

In the DQCS scheme, depicted in Figure 5.11, two queues are used for separately buffering cells from pure and mixed connections. They are referred to as *pure queue* (Queue p) and *mixed queue* (Queue m), respectively. A server attends each queue in cyclic order. The server uses an *exhaustive limited* service policy, where during each visit it serves a maximum of $M_x(x = p, m)$ cells or until the queue is empty, whichever comes first. We assume a zero switchover time between the queues, which means that after serving the current queue, the server switches instantaneously and starts serving the other queue.

5.4.2 Mathematical Modelling

Instead of analysing the scheme directly by formulating joint probabilities of the queue lengths in both queues, we employ an approximated method, in which we view each queue separately as a state-dependent queue with server vacation. In addition to reduction of computational complexity, this gives more flexibility for incorporating an additional buffer threshold in one of the queues and for taking into account the inhomogeneous traffic conditions.

A similar approximation has been previously used for analysing polling systems. Tran-Gia and Raith [1985] presented an iterative algorithm for analysing a polling system with a finite queue and nonexhaustive service discipline (i.e. a special case of exhaustive limited service discipline with $M_x = 1$) with each queue being viewed as a vacation model; Lee [1988a] analysed a vacation model for finite queue and exhaustive limited service discipline for Poisson arrivals; Lang and Bosch [1991] applied the vacation model and the iterative algorithm of [Tran-Gia and Raith, 1985] for analysing a polling system with exhaustive limited service; and Tran-Gia [1992] presented a discrete time analysis of nonexhaustive system with more general input traffic. In the following, we will adopt a similar discrete time analysis technique for evaluating DQCS with exhaustive limited service discipline and Bernoulli input traffic.

For each queue, we define a *polling cycle* as the time interval between two successive arrivals of the server to the queue. It comprises a *service cycle* and a *vacation cycle*, where either cycle can be of zero length. During the service cycle, the queue is served according to an exhaustive limited service discipline, while during the vacation cycle, no cell is served. The length of the vacation cycle of one queue equals the length of the service cycle of the other queue and vice versa. Since there is the possibility that both queues are empty at the same time, we assume that in such situation the server will stay in the current queue until a new cell arrive at either queue. A sample path of a polling cycle with $M_p = 3$ and $M_m = 2$ is shown in Figure 5.12, while the definitions of random variables used in formulating the state probabilities of the queue are listed in the following.

A_x	the number of cells arriving into Queue $x(x = p, m)$ during a slot time.
A_{ml}	the number of low priority cells arriving from mixed connections.
S_x, V_x	the length of the service and vacation cycle for Queue x, respectively.
$Q^-_{xs}(n,k)$, $Q^+_{xs}(n,k)$	the length of queue x immediately prior to, and immediately after, the beginning of the kth slot in the nth service cycle.

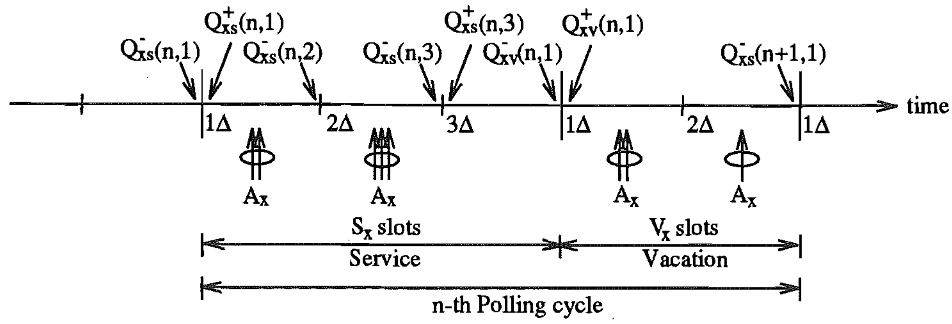


Figure 5.12 Sample path of a polling cycle.

$Q^-_{xv}(n,k)$, $Q^+_{xv}(n,k)$	the lengths of Queue x immediately prior to, and immediately after, the beginning of the kth in the nth vacation cycle.
K_x	the buffer threshold for Queue x.
K_{ml}	the low priority buffer threshold for Queue m.
M_x	the maximum number of cells from queue x that can be served during a service cycle.

Markov Chain State Probabilities

In deriving the state probabilities, the following assumptions are made:

(A5.4) Cell departure takes place at the beginning of a slot.

(A5.5) New cells are admitted into the buffer in batches, at the end of each slot.

(A5.6) The system is in statistical equilibrium.

We first consider the service cycle of one of these two queues. Let us assume that the server is attached to that queue just before the beginning of a time slot, i.e. when the state of the queue is scanned. Depending on this state of the queue, the server will serve the next cell from this queue or take a vacation. It will remain in the service cycle if there is at least one cell waiting in the buffer and the number of cells that have been served during the service cycle is less than M_x. If there is at least one cell in Queue x immediately prior to the beginning of slot k, the state immediately after the beginning of slot k can be given as

$$Q^+_{xs}(n,k) = Q^-_{xs}(n,k) - 1, \quad Q^-_{xs}(n,k) > 0 \tag{5.28}$$

After new arrivals (if any) have joined the queue, its state is given as

$$Q^-_{xs}(n,k+1) = \min(Q^+_{xs}(n,k) + A(k), K_x) \tag{5.29}$$

Letting $\{q^-_{xs}(n,k,j)\}$ and $\{q^+_{xs}(n,k,j)\}$ be the probability mass functions for $Q^-_{xs}(n,k)$ and $Q^+_{xs}(n,k)$, we can write the relationship in (5.28) and (5.29) using the operators Σ and $*$ defined in (3.31)-(3.33), as

$$\begin{array}{lcll} q^+_{xs}(n,k,j) & = & \frac{q^-_{xs}(n,k,j+1)}{1-q^-_{xs}(n,k,0)} & \text{if } 0 \le j \le K_x - 1 \\ q^-_{xs}(n,k+1,j) & = & \Sigma^{K_x}(q^+_{xs}(n,k,j) * a_x(j)) & \text{if } 0 \le j \le K_x \end{array} \tag{5.30}$$

where $a_p(i)$ is defined in (5.1). On the other hand, $a_m(i)$ needs to be conditioned on the queue length j when the cells arrive as in PBS, giving

$$a_m(i|j) = \begin{cases} a_m(i) & \text{if } 0 \le j \le K_{ml} - 1 \\ a_{mh}(i) & \text{if } K_{ml} \le j \le K_m \end{cases} \tag{5.31}$$

where $a_{mh}(i)$ and $a_{ml}(i)$ follow from (5.10) and (5.11)

A similar expression can be written for the vacation cycle, with the exception that no cell is served, hence $Q^+_{xv}(n,k)$ is simply equal to $Q^-_{xv}(n,k)$ and their relationship can be written as

$$\begin{array}{lcll} q^+_{xv}(n,k,j) & = & q^-_{xv}(n,k,j) & \text{if } 0 \le j \le K_x \\ q^-_{xv}(n,k+1,j) & = & \Sigma^{K_x}(q^+_{xv}(n,k,j) * a_x(j)) & \text{if } 0 \le j \le K_x \end{array} \tag{5.32}$$

The transition from a service cycle to a vacation cycle depends on the length of the service cycle, which has a probability mass function $\{s_x(k)\}, k = 0, \ldots, M_x$. The vacation cycle starts with the queue not being empty only when the service cycle is greater than M_x. This condition dictates the state of the queue just prior to the first slot of the vacation cycle $Q^-_{xv}(n,1)$ as

$$q^-_{xv}(n,1,j) = \begin{cases} \sum_{k=0}^{M_x-1} s_x(k) + s_x(M_x)\, q^-_{xs}(n, M_x+1, 0) & \text{if } j = 0 \\ s_x(M_x)\, q^-_{xs}(n, M_x+1, j) & \text{if } 1 \le j \le K_x \end{cases} \tag{5.33}$$

On the other hand, the transition from a vacation cycle to a service cycle in queue x depends on the length of the vacation cycle of the same queue. This in turn depends on the service cycle of the other queue x', which has the maximum duration of M'_x. Letting $\{v_x(k)\}, k = 0, \ldots, M_{x'}$ be the probability mass function of the length of vacation cycle, we can express the state of the queue immediately prior to the first slot of a service cycle as

$$q^-_{xs}(n+1,1,j) = \sum_{k=0}^{M_{x'}} v_x(k)\, q^-_{xv}(n,k+1,j) \tag{5.34}$$

Determination of Service and Vacation Time Distributions

For an exhaustive limited policy, the length of service cycle for queue x equals to k, if the queue is not empty just prior to the beginning of the k-th slot of the service cycle and it becomes empty at the end of the slot or M_x cells have been served if $k = M_x$. The length of service cycle equals to 0 if the server arrives and finds the queue x empty and the other queue x' not empty. This means that the probability mass function of the queue length at the end of kth slot in the nth service cycle, $q^-_{xs}(n,k+1,j)$, determines the probability of the length of the service cycle being k slots, i.e. $s_x(k)$, as

$$s_x(k) = \begin{cases} q^-_{xs}(n,1,0)(1 - q^-_{x'v}(n,1,0)) & \text{if } k = 0 \\ q^-_{xs}(n,k+1,0)(1 - \sum_{i=0}^{k-1} s_x(k)) & \text{if } 1 \le k \le M_x - 1 \\ 1 - \sum_{i=0}^{M_x-1} s_x(k) & \text{if } k = M_x \end{cases} \tag{5.35}$$

Since the distribution of service cycle length for a queue x is equivalent to the distribution of the length of vacation cycle for the other queue x', we have

$$v_x(k) = s_{x'}(k) \tag{5.36}$$

Numerical Algorithm

The dependence of the the length of the vacation cycles for one queue on the length of the service cycles for the other queue, which in turn depends on the steady-state probabilities of the queue lengths, prohibits us from solving the state probabilities directly, hence a numerical algorithm similar to that in [Tran-Gia and Raith, 1985; Lang and Bosch, 1991] will be applied. The algorithm is based on evaluating the distribution of cycle lengths and the steady-state probabilities of queue lengths alternatively. It is detailed as follows.

Algorithm 5.2.

Step 1. Assume the server is at Queue p with one cell in the queue and none in Queue m, i.e. $q^-_{ps}(1,1,1) = 1.0$, $q^-_{mv}(1,1,0) = 1.0$.

Step 2. Calculate the probabilities of queue lengths at the following time slots ($k = 2, \ldots, M_p$) in the service cycle of Queue p and in the vacation cycle of Queue m by using (5.30) and (5.32), respectively.

Step 3. Calculate the distribution of service cycle lengths for Queue p and hence the distribution of vacation cycle lengths for Queue m according to (5.35) and (5.36).

Step 4. Calculate the probabilities of queue lengths for the first time slot, i.e. $k = 1$, in the vacation cycle of Queue p and in the service cycle of Queue m according to (5.33) and (5.34).

Step 5. Calculate the probabilities of queue lengths at the following time slots ($k = 2, \ldots, M_m$) in the vacation cycle for Queue p and in the service cycle for Queue m by using (5.32) and (5.30), respectively.

Step 6. Calculate the distribution of service cycle lengths for Queue m and hence the distribution of vacation cycle lengths for Queue p according to (5.35) and (5.36).

Step 4. Calculate the probabilities of queue lengths for the first time slot, i.e. $k = 1$, in the service cycle of Queue p and in the vacation cycle of Queue m according to (5.33) and (5.34).

Step 8. Repeat steps 2-7 until a convergence criterion is fulfilled.

The convergence criterion is

$$\left| \frac{q^+_{xs}(n,j) - q^+_{xs}(n-1,j)}{q^+_{xs}(n,j)} \right| \leq \epsilon \tag{5.37}$$

where $q^+_{xs}(n,j) = \sum_{k=0}^{M_x} q^+_{xs}(n,k,j)$ and by default $\epsilon = 1 \times 10^{-6}$ and $\lim_{n\to\infty} q^+_{xs}(n,k,j) = q^+_{xs}(k,j)$ when steady-state is reached.

Calculation of Cell Loss Probabilities

After calculating the steady-state probabilities of queue lengths, we can proceed to calculate the cell loss probabilities. In deriving these formulas we distinguish between losses in the service cycle and in the vacation cycle. During service cycle of Queue p, we observe any arbitrary cell within an arrival group of i. Since the cells arrive after a cell in the queue (if any) has commenced its service, based on assumptions (A5.4) and (A5.5), we can only accept up to K_p cells. Hence, given that the queue is in state j, cell loss will occur if $j+i > K_p$, and $i-(K_p-j)$ of the arriving cells will be rejected. The average number of cells lost is given by

$$L_{ps}(k,j) = \sum_{i=K_p-j+1}^{N_p} [i-(K_p-j)]\, a_p(i) \tag{5.38}$$

Unconditioning over k and j, we have

$$L_{ps} = \sum_{k=1}^{M_p} s_p(k) \sum_{j=0}^{K_p-1} q_{ps}^{+}(k,j)\, L_{ps}(k,j) \tag{5.39}$$

We can apply the same arguments to the vacation cycle, except that no cell is served during this cycle. This leads to

$$L_{pv} = \sum_{k=1}^{M_m} v_p(k) \sum_{j=0}^{K_p} q_{pv}^{+}(k,j) \sum_{i=K_p-j+1}^{N_p} [i-(K_p-j)]\, a_p(i) \tag{5.40}$$

By taking the average cell losses in both cycles, we obtain an average number of cells lost per slot. Dividing this result by the average number of cells arriving per slot, we obtain the cell loss probability as

$$P_{p(DQCS)} = \frac{L_{ps}E[S_p] + L_{pv}E[V_p]}{(E[S_p]+E[V_p])\,\rho_p} \tag{5.41}$$

where $E[S_p]$ and $E[V_p]$ are the average length of service cycle and vacation cycle, respectively.

Similar derivations are used for Queue m which result in the average losses of high priority cells L_{msh}, L_{mvh} and low priority cells L_{msl}, L_{mvl} in the service cycle and vacation cycle being

$$L_{msh} = \sum_{k=1}^{M_m} s_m(k) \sum_{j=0}^{K_m-1} q_{ms}^{+}(k,j)\, L_{msh}(k,j) \tag{5.42}$$

$$L_{msl} = \sum_{k=1}^{M_m} s_m(k) \sum_{j=0}^{K_m-1} q_{ms}^{+}(k,j)\, L_{msl}(k,j) \tag{5.43}$$

$$L_{mvh} = \sum_{k=1}^{M_p} v_m(k) \sum_{j=0}^{K_m} q_{mv}^{+}(k,j)\, L_{mvh}(k,j) \tag{5.44}$$

$$L_{mvl} = \sum_{k=1}^{M_p} v_m(k) \sum_{j=0}^{K_m} q_{mv}^{+}(k,j)\, L_{mvl}(k,j) \tag{5.45}$$

where

$$L_{msh}(k,j) = \begin{cases} \eta_m \sum_{i=K_m-j+1}^{N_m} [i-(K_m-j)]\, a_m(i) & \text{if } 0 \le j \le K_{ml}-1 \\ \sum_{i=K_m-j+1}^{N_m} [i-(K_m-j)]\, a_{mh}(i) & \text{if } K_{ml} \le j \le K_m - 1 \end{cases} \tag{5.46}$$

$$L_{msl}(k,j) = \begin{cases} (1-\eta_m)\sum_{i=K_m-j+1}^{N_m}[i-(K_m-j)]\,a_m(i) & \text{if } 0 \le j \le K_{ml}-1 \\ \sum_{i=1}^{N_m} i\,a_{ml}(i) & \text{if } K_{ml} \le j \le K_m - 1 \end{cases} \tag{5.47}$$

$$L_{mvh}(k,j) = \begin{cases} \eta_m \sum_{i=K_m-j+1}^{N_m}[i-(K_m-j)]\,a_m(i) & \text{if } 0 \le j \le K_{ml}-1 \\ \sum_{i=K_m-j+1}^{N_m}[i-(K_m-j)]\,a_{mh}(i) & \text{if } K_{ml} \le j \le K_m \end{cases} \tag{5.48}$$

$$L_{mvl}(k,j) = \begin{cases} (1-\eta_m)\sum_{i=K_m-j+1}^{N_m}[i-(K_m-j)]\,a_m(i) & \text{if } 0 \le j \le K_{ml}-1 \\ \sum_{i=1}^{N_m} i\,a_{ml}(i) & \text{if } K_{ml} \le j \le K_m \end{cases} \tag{5.49}$$

The overall loss probabilities are

$$P_{mh(DQCS)} = \frac{L_{msh}E[S_m] + L_{mvh}E[V_m]}{(E[S_m]+E[V_m])\rho_{mh}} \tag{5.50}$$

$$P_{ml(DQCS)} = \frac{L_{msl}E[S_m] + L_{mvl}E[V_m]}{(E[S_m]+E[V_m])\rho_{ml}} \tag{5.51}$$

5.4.3 Analysis Verification

In this section, we will verify the analytical results obtained for the DQCS with the results obtained from simulation by considering an ATM switch of size 16 $\times$ 16. We assume that there are equal numbers of pure and mixed connections and that they offer equal loads. The ratio of high priority traffic to total traffic of mixed connections is 0.5.

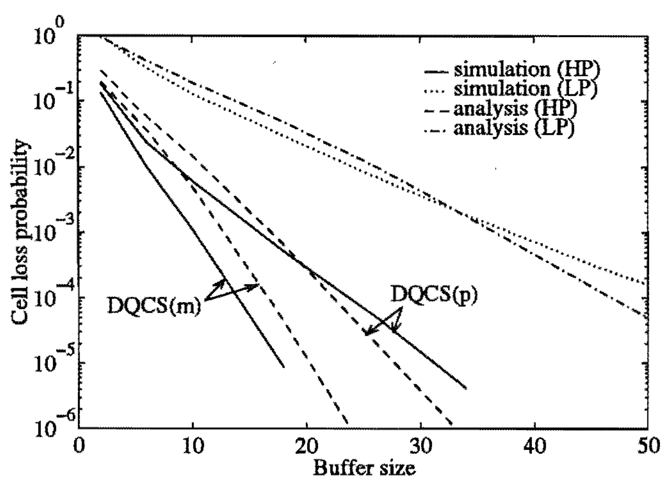


Figure 5.13 Cell loss probability versus buffer sizes with $\rho_p = 0.4$, $\rho_m = 0.4$ and $\eta_m = 0.5$.

In Figure 5.13 we compare the cell loss probability for varying buffer sizes with pure and mixed traffic offering an equal load of 0.4, i.e. $\rho_p = \rho_m = 0.4$, and $M_p = M_m = 8$. Evidently the analytical results for traffic of pure connections and low priority traffic of mixed connections are initially higher than that of simulation. As the buffer size increases, the loss probabilities eventually drop below that of simulation.

In Figure 5.14 we compare the cell loss probability for varying offered load with $K_p = K_m = 4$ and $M_p = M_m = 8$. The analytical results show a reasonable agreement to the simulation results.

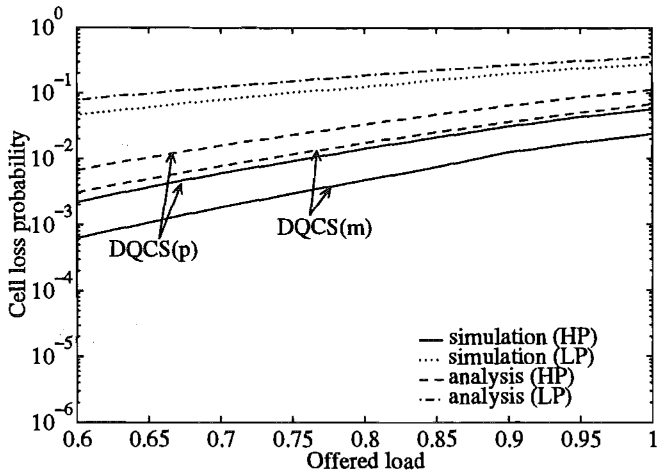


Figure 5.14 Cell loss probability versus offered load with $K = 8$.

The agreement tends to improve as the offered load increases. For example, when $\rho_p + \rho_m = 0.6$, the analytical results for traffic of pure connections are three times higher than the simulation results, while they are only two times higher than the simulation results when $\rho_p + \rho_m = 1.0$.

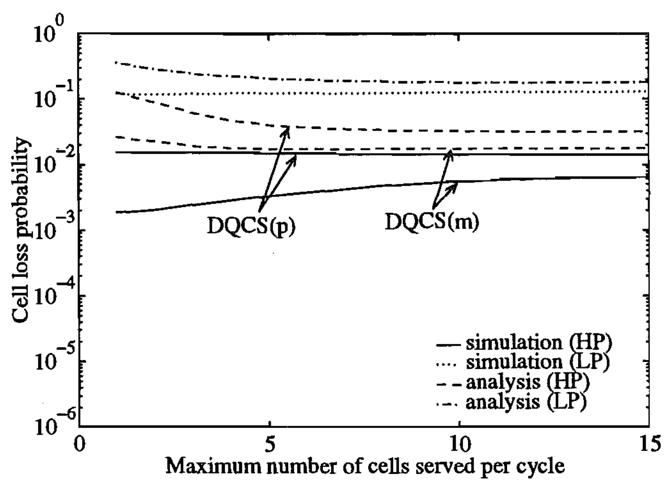


Figure 5.15 Cell loss probability versus maximum number of cells served per visit.

In Figure 5.15 we compare the cell loss probabilities for varying maximum number of cells served per visit (M_x) when $\rho_p = \rho_m = 0.4$ and $K_p = K_m = 4$. Notice that the analytical results demonstrate reasonable agreements to the simulation results and these agreements improve as M_x increases.

Overall, we found that the analytical results provides reasonable agreements to the simulation results only for a certain range of values. Although the results can be used as a rough indication of the DQCS performance, they are not suitable for comparison study as the level of inaccuracies is too high. For this reason, simulation studies will be used for comparing the performance of the

DQCS with other schemes, as discussed in Section 5.5.5. Further work is obviously needed to improve the DQCS analysis, possibly by taking into account the joint queue lengths between the queues.

5.5 Improved Shared Buffer Policies

In this section, we will discuss two recent improvements to the classical PBS and PO schemes and two recent proposals for simulation-based protective buffer policies. All of these schemes appeared after the original proposal of the DQCS in [Hartanto *et al.*, 1991]. The performance comparison among the schemes is carried out based on simulation. We also compare these schemes with other policies discussed in previous sections, namely with PO, PBS, RS and DQCS.

5.5.1 Limited Push-out Scheme

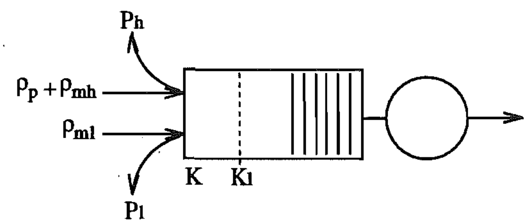


Figure 5.16 Limited push-out (LPO) scheme.

The *limited push-out (LPO)* scheme was proposed by Beraldi and Maranao [1992] for reducing the complexity of PO by limiting the number of buffer places from where low priority cells can be pushed out. It can also be viewed as an improvement to PBS scheme that increases the admissible low priority traffic by allowing low priority cells to be queued in the part of buffer reserved for high priority cells.

Under LPO, a buffer of size K is divided in two parts as depicted in Figure 5.16. The first part, which includes last $K - K_l$ places, is a risk area for the low priority cells because they could be pushed out by high priority cells finding the buffer full. The replacement strategy is LIFO. In the second part all cells are handled in the same manner, regardless of their priority, as in a normal FIFO queue. Low priority cells in this part can not be pushed out. The analysis for the scheme with Bernoulli input traffic is given in [Beraldi and Maranao, 1992].

5.5.2 Threshold Push-out Scheme

The *threshold push-out (TPO)* scheme is proposed by Tipper *et al.* [1993] to offer some adjustment to the cell loss rates of various priority classes in a push-out scheme. This is done by introducing a set of overwrite thresholds for all priority classes.

Under TPO with high and low priority classes of cells, the set of overwrite thresholds will be $[K_h, K_l]$, where the sum of the thresholds equals the total buffer spaces (i.e. $K_h + K_l = K$). In the scheme, all cells are admitted until the common buffer space is full. A high priority cell arriving at a full buffer can overwrite a low priority cell if the number of low priority cells in the buffer exceeds the K_l threshold, otherwise the cell is discarded. Similarly, a low priority cell arriving at a full buffer can push out a high priority cell if the number of high priority cells in the buffer exceeds the K_h threshold, otherwise the cell is discarded. The analysis of that scheme, assuming a Bernoulli input model, can be found in [Tipper *et al.*, 1993; Suri *et al.*, 1994].

5.5.3 Simulated Protective Policy

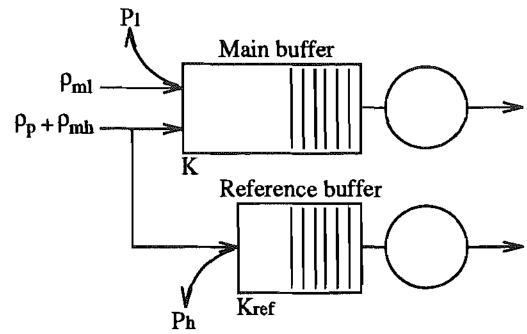


Figure 5.17 Simulated protective policy (SPP).

The *simulated protective policy (SPP)*, proposed by Cidon *et al.* [1993], attempts to enhance the service of low priority traffic through limiting the service of high priority traffic to the minimum required from a policy to be protective.

As depicted in Figure 5.17, SPP uses two buffers: *main buffer* and *reference buffer*. The size K_{ref} of reference buffer is chosen to satisfy QoS of high priority traffic. The reference buffer is simulated in parallel to the operations of the main buffer. Both buffers operate at the same service rate.

SPP ensures that the main buffer does not discard more high priority cells than the reference buffer by requiring the main buffer to have more space free for future arrivals of high priority cells than the reference buffer. Let Q_{main} and Q_{ref} denote the current number of high priority cells in the main and the reference buffer, respectively. SPP ensures that the policy remains protective by keeping $(K - Q_{main}) > (K_{ref} - Q_{ref})$ at all times. The buffer space occupied by low priority cells, except the one in service, is counted as free since the low priority cells can be pushed out at any time to make room for arriving high priority cells.

The operation of SPP is based on the following rules. An arriving high priority cell will be accepted into the main buffer if the reference buffer is not full and it can find a free space in the main buffer, or if it can push a low priority cell out of the main buffer, otherwise it will be discarded. The accepted high priority cell will be duplicated in the reference buffer. The discarding of high priority cells rejected by the reference buffer leaves more space in the main buffer for low priority cells. An arriving low priority cell will always be accepted as long as the main buffer is not full. Low priority cells are replaced in the LIFO order.

When the main buffer is about to serve a low priority cell, it has to make sure that it has at least one more buffer place available for high priority cells than the reference buffer. If this is not the case, then the low priority cell is dropped and any low priority cells following that cell will also be dropped, since the same rejection rule would apply to each of them. The service will then be given to the first high priority cell encountered. Regardless of the type of cells served by the main buffer, the reference buffer will always serve a cell as long as it is not empty.

In practice, the reference buffer can be replaced by a counter with its current value representing Q_{ref}. The counter is incremented whenever a high priority cell arrives and is accepted by the main buffer provided the counter value is lower than K_{ref}. On the other hand, the counter is decremented everytime a cell in the main buffer is served and the counter value is greater than 0.

5.5.4 Extended Simulated Protective Policy

Under SPP, a low priority cell, which is about to be served, may be discarded if the free space in the main buffer is smaller than in the reference buffer. The *extended simulated protective policy (ESPP)*, proposed by Cidon *et al.* [1993], attempts to enhance SPP by identifying such cells as early as possible, so the buffer space freed due to the discarding of these cells can be used by other arriving cells. To identify these cells, ESPP takes the following additional action when a high priority cell is accepted into the main buffer. It searches for the first low priority cell in the main buffer and checks whether the cell is doomed to be dropped later. The cell is identified as doomed if $(K - Q_{main} - 1) + X < \min(K_{ref} - Q_{ref} + X, K_{ref})$, where X is the number of high priority cells preceeding the cell. If the cell is doomed, then it will be discarded and any low priority cells following the cell will be discarded as well. The process moves on to check the next low priority cell and continues to do so until all low priority cells have been discarded or until a low priority cell, which is not doomed, is found.

5.5.5 Performance Comparisons

To compare performance of all buffer policies presented in the previous section, we have conducted simulation studies under the same assumptions as those in [Cidon *et al.*, 1993], i.e. we choose $QoS_p = QoS_{mh} = 10^{-3}$ and $QoS_{ml} = 10^{-1}$. On the other hand, contrary to [Cidon *et al.*, 1993] the final simulation results were obtained with the required precision at a given confidence interval, following the method of sequential quantitative simulation described in [Hartanto *et al.*, 1994a]. Assuming $\rho_{p(max)} = \rho_{mh(max)} = 0.4$ and $\rho_{ml(max)} = 0.1$, we can find the required buffer thresholds as shown in Table 5.2.

	PO	PBS	LPO	TPO		SPP	ESPP		RS	DQCS
K	16	14	17	18	K	16	16	K_p	11	11
K_l	-	8	7	1	K_{ref}	11	11	K_{mh}	27	11
								K_{ml}	20	5
Total	16	14	17	18		16	16		38	22

Table 5.2 Optimum buffer thresholds (QoS_p=QoS_{mh}=10^{-3}, QoS_{ml}=10^{-1}).

We notice that a separate buffer policy requires more buffers than a shared buffer policy. PBS requires the least buffer space, while RS requires the largest buffer space of all schemes. DQCS allows a 40% reduction of the required buffer space by RS. The buffer sizes for LPO and TPO are larger than PO. This is due to the limitation enforced on the high priority cells in pushing out low priority cells in LPO and due to the possibility of low priority cells pushing out high priority cells in TPO.

Admissible Levels of Low Priority Traffic

In this study, we assume that the buffer thresholds of all policies are fixed. Due to the small differences between the admissible traffic among the buffer policies, we will tabulate them in Table 5.3 for $\rho_p = 0.20, \rho_{mh} = 0.40$ and $\rho_p = \rho_{mh} = 0.35$. The results are obtained by simulation and they have a precision below 0.05 at the 95% confidence level.

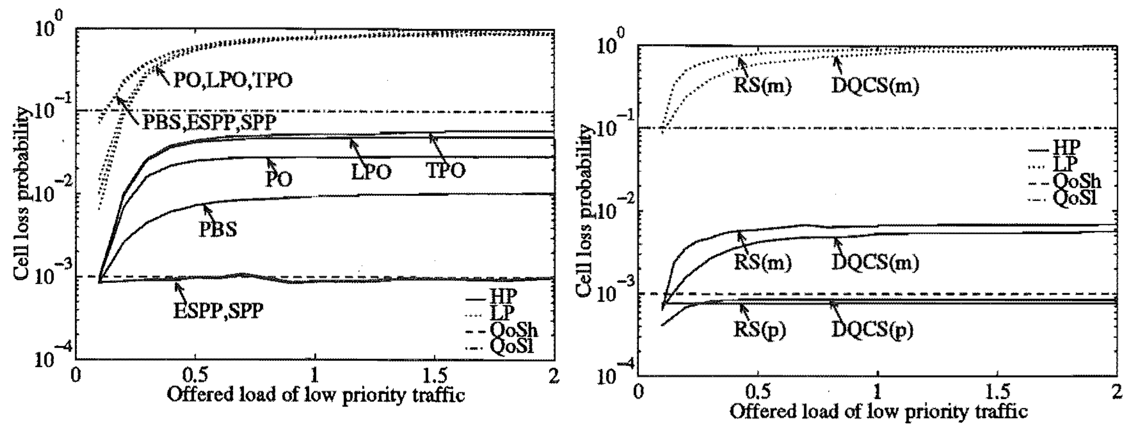
Offered load	PBS	PO	LPO	TPO	SPP	ESPP	RS	DQCS
$\rho_p = 0.20, \rho_{mh} = 0.40$	0.383	0.400	0.364	0.383	0.378	0.373	0.100	0.328
$\rho_p = 0.35, \rho_{mh} = 0.35$	0.254	0.241	0.222	0.229	0.246	0.239	0.150	0.223

Table 5.3 Admissible low priority traffic (QoS_p=QoS_{mh}=10^{-3}, QoS_{ml}=10^{-1}).

The admissible level of low priority traffic under PO, LPO and TPO is limited by the QoS of high priority traffic, while in the remaining schemes it is limited by the QoS of low priority traffic. As shown in the table, PO can admit the most low priority cells when $\rho_p + \rho_{mh} = 0.6$ because of its complete sharing of bandwidth and buffer resources between both classes of traffic. On the other hand, as expected RS admits the fewest low priority cells. LPO and TPO admit fewer low priority cells than PO because of the restriction put on the high priority cells in pushing out low priority cells, hence the degradation of QoS for high priority traffic is faster than under PO. DQCS admits more than three times low priority cells than RS due to the sharing of idle bandwidths from class of pure connections.

One peculiar observation from the table is that SPP admits more low priority cells than ESPP, which is in contradiction to the results obtained in [Cidon *et al.*, 1993]. A possible explaination for this contradiction is as follows. In proposing ESPP, Cidon *et al.* [1993] assume that discarding low priority cells as early as possible will allow other low priority cells, which have better chances to be served, to enter the buffer. However, it is possible that the cells may be discarded unnecessarily while there is actually capacity to accomodate them, because no other low priority cells arrive to replenish the buffer. Following that argument, it means that the wrong conclusion has been drawn in [Cidon *et al.*, 1993] and this was probably because of their simulation results were obtained without any consideration of the precision of the results. As illustrated in [Hartanto *et al.*, 1994a], such a simulation practice can easily lead to a wrong conclusion. In order to avoid such pitfalls, proper statistical analysis of simulation output data is an utmost necessity, especially in comparing schemes with very small differences between the loss probabilities.

Protection Levels of High Priority Traffic Against Overload of Low Priority Traffic



(a) Shared buffer policies

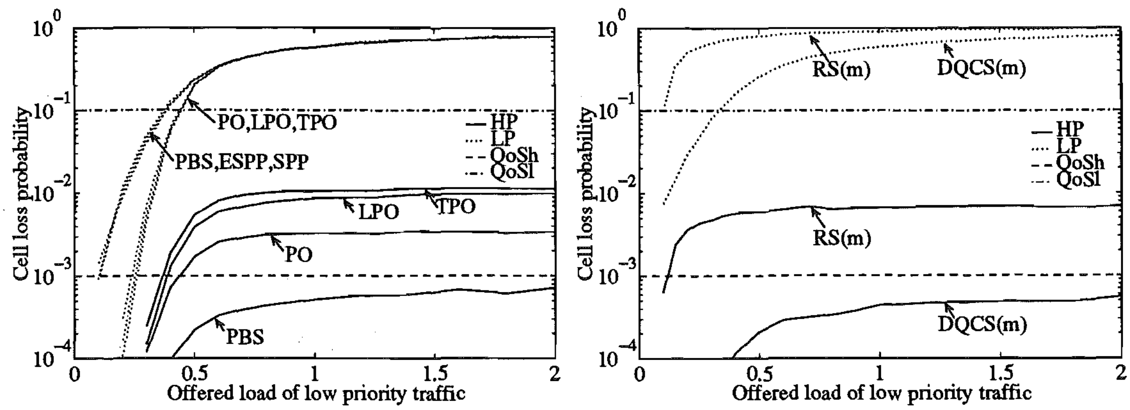
(b) Separate buffer policies

Figure 5.18 Cell loss probability for various offered load of low priority traffic with $\rho_p = 0.4$, $\rho_{mh} = 0.4$

Figure 5.18(a) shows that SPP and ESPP are able to keep the loss probability of high priority cells below the QoS requirements (the statistical fluctuation is due to the simulation process), while the loss probability under other schemes increases as the level of overload increases. As expected TPO, which allows low priority cells to push high priority cells out of the buffer, offers the worst quality of service to high priority traffic, while treating low priority traffic in the best way. Since no adjustment of the buffer threshold can be made to reduce the loss probability for high priority traffic below QoS_p or QoS_{mh}, this means that TPO policy is non-protective. LPO, which limits the buffer space from where low priority cells can be pushed out, also performs worse than PO. The best performance achievable when $K_l = 0$ is just equal to the performance of PO and is still worse than the required QoS_p, For this reason, we can conclude that LPO policy is non-protective. On the other, PBS allows to adjust the buffer threshold in order to satisfy the required QoS_p. This again confirms that PBS policy is protective.

Figure 5.18(b) shows the protection levels of separate buffer policies. As expected, under RS, the performance of traffic of pure connections (indicated by RS(p) in Figure 5.18(b)) is not affected by the overload. It is affected in DQCS, which shares some resources among the traffic classes. However, the cell loss probability under DQCS stays below the required QoS_p. Due to the resource sharing, the performance of traffic of mixed connections under DQCS is better than that under RS. The loss probability for the high priority traffic of mixed connections is higher than the required QoS_{mh}. The network can leave the loss probability as it is, in order to deter the users from using cell tagging too excessively, or it can adjust the buffer threshold used in DQCS to bring the loss probability below the required QoS_{mh}. The same thing can be said about RS, where by adjusting K_{ml} to 16, we obtain a loss probability below QoS_{mh} when $\lambda_{ml} = 2.0$. This indicates that both DQCS and RS policies are protective. As compared to PBS, DQCS has the advantage of always ensuring that the QoS for traffic of pure connections is satisfied.

The graphs in Figure 5.19(a) show that LPO, TPO and PO can not satisfy the high priority QoS and therefore confirm that the buffer policies are non-protective. On the other hand, the loss



(a) Shared buffer policies

(b) Separate buffer policies

Figure 5.19 Cell loss probability for various offered load of low priority traffic with $\rho_p = 0.2$, $\rho_{mh} = 0.4$

probability under PBS, SPP, and ESPP falls well below the required QoS. This again confirms that the buffer policies are protective. SPP and ESPP have been proposed to improve the performance of low priority traffic by just satisfying the high priority QoS. Thus, by adjusting K_{ref} to 6 when $\lambda_{ml} = 2.0$, a smaller loss probability for low priority traffic can be obtained. However, this probability is still higher than that under TPO, which offers the best performance among the shared buffer policies.

For separate buffer policies, Figure 5.19(b) shows that the loss probability for high priority traffic of mixed connections is unchanged under RS, but lower than the required QoS_{mh} under DQCS. This again demonstrates the benefit of a dynamic sharing of bandwidth between both traffic classes in DQCS. Overall the results confirm that RS and DQCS policies are protective.

In summary, we can conclude that DQCS, SPP and ESPP are the best protective policies for offering full protection to traffic of pure connections against the overload of low priority traffic of mixed connections. DQCS offers additional advantage of controlling the loss probability of traffic of pure connections and high priority traffic of mixed connections separately. This allows a possibility for the network to let the overload by low priority traffic have some effect on the performance of high priority traffic of mixed connections in order to deter excessive use of cell tagging by users.

5.6 Conclusion

In Section 5.2.2, we have defined a more general criterion for evaluating protective levels of a buffer policy than that stated in [Cidon *et al.*, 1993]. Based on this criterion, we have investigated four classical buffer management schemes and five recently proposed schemes. Among the classical schemes, we found PBS and RS to be protective, while CBS and PO to be non-protective. The protection offered by RS has been solely achieved through dedicating resources to the class of pure connections. Such reservation leads to a waste of bandwidth resources when the offered load of the class is low.

In order to improve the performance of RS, we have proposed a new scheme, DQCS, which allows some sharing of the idle bandwidth. Performance comparison study has shown that such sharing of bandwidth allows DQCS to offer a better performance than RS.

Performance comparisons between DQCS, PO, PBS and other recently proposed schemes have also been carried out by using simulation. The results reveal that the performance of DQCS is similar to that under SPP and ESPP, which have been proposed recently by Cidon *et al.* [1993]. The other two recently proposed schemes, LPO and TPO, have been found to be non-protective. Their protection levels are at most the same as that of PO.

With either DQCS, SPP or ESPP offering full protection to traffic of pure connections irrespectively of the amount of overload by low priority traffic, it could be possible to realise the suggestion at the end of Chapter 4, i.e. for abandoning policing low priority traffic by the network in order to maximise the network utilisation. The possible degradation of high priority traffic of mixed connections under DQCS when the traffic is at maximum load can be used by the network to deter excessive use of cell tagging by users.

Chapter 6

CONCLUSIONS AND SUGGESTIONS FOR FUTURE WORK

In this final chapter, we review the main contributions in this thesis and give some suggestions for extending the research.

6.1 Conclusions

Recent standardisation of ATM technologies has offered an enormous potential for building B-ISDNs. However, additional control functionalities are required at layers above the ATM layer in order to provide useful services. This thesis has examined two key requirements for implementing B-ISDN services, namely call control and traffic management. A review of the state of the art of these two aspects along with the recent standardisation in ATM technologies is presented in Chapter 1.

The call control aspect was studied in Chapter 2, where we introduced a user-network oriented call control for managing calls with multiple parties and multiple connections. The proposed control architecture has a layered structure and incorporates an additional party control layer. By defining protocol commands for each layer, we have demonstrated its benefits in establishing complex calls as well as simple calls. The control structure offers flexibility to complex calls and allows progressive advancement of the users' terminals in making use of B-ISDN signalling capabilities. The control structure also allows simple calls to bypass the higher control layer. Such bypassing has the potential for reducing the call establishment delay for this type of call. In order to facilitate multipoint connections, we have proposed some additional capabilities for a multicast switch and simulation studies have shown that these additional capabilities can also help in reducing contention within the multicast switch.

In the traffic management aspect, we have focused on the applications of cell loss priority by users as opposed to by the network. Based on the absence or existence of pretagged or low priority cells within a connection, we have classified connections into two classes, i.e. pure and mixed. The traffic management framework for mixed connections has been developed in Chapters 3-5, which includes connection admission control, usage parameter control and buffer management systems. In analysing each component of the framework, we have made use of a unified discrete-time analysis and simulations, using DESC++ package [Hartanto *et al.*, 1994a], have been carried out to verify the analysis.

Chapter 3 was devoted to the development of bandwidth allocation methods for mixed connections, which involved traffic modelling and multiplexer analysis. We modelled each connection by an IPP source. The superposition of the IPP sources was approximated by an MMPP source. In order to match the traffic characteristics of the superposed sources to that of the MMPP source, we investigated three recently proposed matching methods, namely the LL method [Lee and Lee, 1992], the BBLRW method [Baiocchi *et al.*, 1991] and the WS method [Wang and Silvester, 1993]. Comparison of the matching results indicated that the WS method offers the best matching procedure, since it can model the overload in the actual system better than the other methods. Modelling the input traffic as an MMPP source, we presented a discrete-time analysis of a multiplexer without and with priority management. We used the analysis for investigating six bandwidth allocation methods in a homogeneous environment and two bandwidth allocation methods in a heterogeneous environment. The results demonstrate the advantages of pretagging of cells by users in reducing the bandwidth requirement. The methods proposed in this thesis, Methods V and VI for a homogeneous environment and the AT approach for a heterogeneous environment, have been shown to allocate the least amount of bandwidth while satisfying the QoS requirement. The benefit for the network is that it can support more connections of the same kind, hence increasing the level of resource utilisation.

Chapter 4 was devoted to the development of leaky bucket schemes for policing mixed connections. Considering the large number of existing leaky bucket schemes, we proposed a classification on the basis of the number of token pools (single and dual leaky bucket) and of the treatment of pretagged cells in a mixed connection (non-priority and priority types of leaky bucket). Following the classification, we presented all possible combinations of three policing mechanisms, namely discarding, buffering and marking. Four of the schemes, DLBP, BDLBP, BDLBP-DT and BDLBMP, were uncovered as a result of the classification. We analysed these schemes on a discrete-time basis along with the other known schemes considered in this Chapter. The analytical results were compared with simulation results; A reasonable agreement was found, the difference is less than 10% in most cases. Comparative studies of the performance offered by all schemes considered were carried out by means of simulation techniques. The loss probabilities for untagged and tagged traffic were used as performance measures for leaky buckets policing the average rate of video sources generating pretagged traffic. In general, the results showed that the priority schemes outperformed the non-priority ones. Comparison between the priority schemes with and without cell marking by the network reveals very little difference in the performance. These results support our proposal for cell marking or tagging to be done by the users and not by the network in order to allow users to have more control over which cells may be discarded. In order to maximise network utilisation, we have proposed a possibility for the network to abandon policing low priority traffic from users. However, with this proposal, there is a need for the network to implement a buffer management policy which can protect traffic of pure connections from overload by low priority traffic.

The issue of protecting traffic of pure connections was studied in Chapter 5, where we defined a protection criterion. This criterion is more general than that stated in [Cidon *et al.*, 1993] and allows comparisons among various buffer management schemes which use not only shared buffer policies but also separate buffer policies. Based on the criterion, we analysed and compared the levels of protection and the admissible levels of low priority traffic under four classical buffer management schemes (CBS, PBS, PO and RS) and showed that PBS and RS policies are protective. RS offers a better solution than PBS in protecting traffic of pure connections since it does not require adjustment of the buffer threshold. However, it admits fewer low priority cells than PBS. In order to overcome this drawback of RS, we proposed and analysed DQCS on a discrete-time basis. We compared the performance of DQCS, PBS, PO, RS and four other buffer management schemes, which have been proposed after the initial proposal of DQCS in [Hartanto *et al.*, 1991], by means of simulation. The results showed that DQCS, SPP and ESPP provided the best protection for traffic of pure connections, even when the traffic of pure connections and the high priority traffic of mixed connections is at their maximum load. DQCS offers the additional advantage of controlling the loss priority of traffic of pure connections and high priority traffic of mixed connections separately. This allows the possibility for the network to let the overload of low priority traffic have some effect on the performance of high priority traffic of mixed connections in order to deter excessive use of cell tagging by users and to force users to act more responsibly.

In summary, we have advocated greater user involvement in the control process in this thesis, and a control framework based on this approach has been proposed and its benefits as compared to the status quo of network based control have been demonstrated. We have also advocated wider use of cell loss priority by users in order to take advantage of user knowledge about their traffic. Conversely, we discourage its usage by the network as the network does not know the significance of information within ATM cells. The network should only police the traffic from the users without carrying out any cell marking, for example by using DLBP or BDLB, and should guarantee transport of the traffic that has entered the network, especially the traffic of pure connections, by using for example DQCS. On the other hand, the network can also prevent users from exploiting the freedom of using cell tagging, for example, by allowing the overload of low priority traffic to degrade the QoS of high priority traffic of mixed class connections. To do so, again DQCS would be an excellent control tool.

6.2 Future Work

The control framework presented in this thesis represents only a step towards developing an overall user-network oriented call control and traffic management framework. There are many problems that have not been sufficiently addressed. One of such problems, mentioned in Section 2.3.1, is related to the synchronisation of connections. The problem is outlined in Section 6.2.1 with a review of known solutions to the problem and a proposed solution for further study. Another problem relates to the control of data services in ATM networks, where additional control is required to complement the traffic management framework discussed in this thesis in order to resolve specific service-related problems. In recent years, burst-level control schemes have often been suggested to resolve the problems. However, as will be discussed in Section 6.2.2, such schemes cannot efficiently cater for the whole range of burst lengths in data services. Therefore the possibility of combining cell-level and burst-level control to provide an integrated control for data services is discussed in Section 6.2.2. This integrated control for data services promises further potential for providing an overall traffic management based on the user-network oriented approach where users have the flexibility to choose which type of control technique to use depending on the services required while the control techniques employed by the network are more uniform. This control framework is discussed in Section 6.2.3.

6.2.1 Synchronisation of Connections

The customer call control structure proposed in Chapter 2 allows users to deliver various media streams from multimedia applications by using separate connections. The media can be time-dependent (e.g. audio, video) or time-independent (e.g. text, graphics, images). Some of the media streams have temporal relationships among one another. For example, in a videophone call a video stream (e.g. the speaker's face and lips movement) has to be synchronised with the corresponding voice. Having the media streams traversing different paths of variable delay characteristics, the time characteristics of the media units (e.g. voice segments or cells) may be distorted when they arrive at the receiver as illustrated in Figure 6.1. The distortation can be in the form of jitter and gaps within a single connection, and skew between two connections.

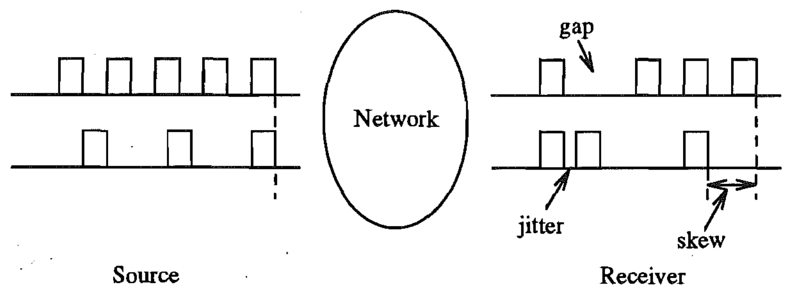


Figure 6.1 Synchronisation problems.

The process of recovering or maintaining the temporal order of media units at the receiver end is referred to as synchronisation. Steinmetz [1990] overviews the fundamental issues in performing synchronisation. Two basic synchronisation requirements can be identified as *intermedia synchronisation* to maintain the temporal coordination of multiple streams relative to one another, and to play out concurrent streams in synchronism so as to avoid skewing, and *intramedia synchronisation*, which requires each media unit within an individual stream to be delivered without loss and with the same delay so as to avoid gaps and jitter.

Intramedia synchronisation has been widely studied based on either buffer filling or time stamp methods [De Prycker, 1991], whereas much work remains to be carried out on intermedia synchronisation. Intermedia synchronisation for point-to-point multimedia applications can be resolved by using multimedia virtual circuit (MVC) concept [Leung *et al.*, 1990], or by multiplexing all media streams into a single connection and inserting intersynchronisation information [Nicolaou, 1990]. Other recently proposed methods include a segment constraint control [Li *et al.*, 1992], which controls the segmentation of voice and video based on the voice on-off characteristics. The drawback of the abovementioned methods is that they require the streams to originate from the same source.

In recent years, Little and Ghafoor [1991b] has proposed a mechanism for synchronising multiple media streams originating from several independent sources distributed throughout the network. However, the mechanism requires that the characteristics of the media streams be known in advance. A more general synchronisation protocol has been proposed by Escobar *et al.* [1992]. In the protocol, a media unit is time stamped with the network clock at the source.

Upon receiving the media unit, the receiver will calculate its playout time based on the time stamp plus a control time. The control time is the same for all connections and is chosen to allow for the worst communication delay. The drawback of this method is that excessive delays in one connection will cause all connections to be delayed excessively. To overcome the drawback, Naish [1994] has proposed an intermedia synchronisation method based on a combination of time stamps and synchronisation markers. For isochronous media, synchronisation markers are used to maintain the instantaneous delay variation in playout whereas time-stamps are used for facilitating intermedia synchronisation. For anisochronous media, time-stamping is used for both types of synchronisation. The drawback of this method is that it requires source clock and receiver clock to be synchronised to the network clock. This requirement tends to conflict with the asynchronous nature of ATM networks.

In order to avoid the strong dependency on network clock synchronisation and also to provide a single synchronisation protocol for isochronous and anisochronous media streams, we propose the possibility of using independent source clock and receiver clock that have the same clock modulus (compare modulus of ordinary clock is 12 or 24 hours). This method allows synchronisation to a single receiver clock rather than to a network clock or a single source clock. The basic idea of this protocol and its two major components, namely *pilot stamp* and *offset stamp*, are shown in Figure 6.2.

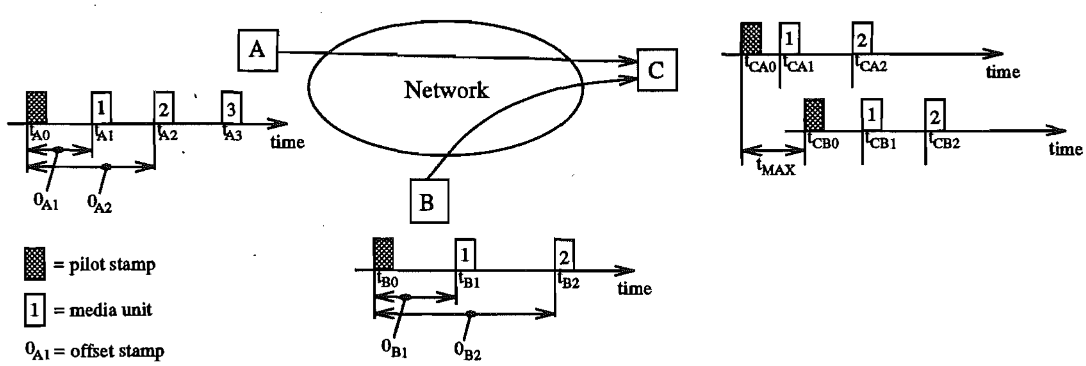


Figure 6.2 Synchronisation protocol components.

Each source will insert a pilot stamp at the beginning of each stream. The pilot stamp is used to provide the initial temporal relationship between the source clock and the receiver clock. The use of a pilot stamp instead of the first media unit for this purpose is to ensure that the pilot stamp will not be discarded within the network. It can be sent through the signalling channel (outband signalling) or through the user connection (inband signalling). Upon receiving the pilot stamp, the receiver will note its own clock, say t_{CA0} for stream A. This clock can be adjusted, say t'_{CA0} to indicate the actual starting time of stream A. If stream A has to be played out simultaneously with stream B, then the receiver will have to wait at most t_{MAX} before playing out stream A. If by that time, no pilot stamp is received for stream B, then the receiver can start playing out the first media unit received for the stream A. This implies that $t'_{CA0} = t_{CA0} + t_{MAX}$ if no pilot stamp for stream B is received and $t'_{CA0} = t_{CB0}$ otherwise, i.e. $t'_{CA0} = \min(t_{CA0} + t_{MAX}, t_{CB0})$. This procedure basically implements the intermedia synchronisation.

An offset stamp, which indicates the time offset from the last pilot stamp, is encoded into each media unit. Upon receiving the media unit, the receiver will calculate the playout time for each media unit as $t'_{CA0} + O_{A1}, t'_{CA0} + O_{A2}, \ldots$. These offset stamps basically facilitate the intramedia synchronisation.

Prior to evaluating the performance of the schemes, either by simulation or practical experiments, there are some issues which should be considered including the size of offset stamp and the clock modulus, the effects of late and lost cells which cause gaps on the synchronisation, and possible clock drift between the source and receiver.

6.2.2 Combined Cell-Level and Burst-Level Control

As mentioned in Section 1.6, CCITT has defined a family of traffic management mechanisms, which includes connection admission control (CAC), usage parameter control (UPC), buffer management and scheduling (BMS), and reactive control. In general, the framework is sufficient for voice and video services, especially if the additional capabilities for handling pretagged cells are included, as discussed in Chapters 3-5. However, for data services, the traffic management framework should be complemented by other types of control in order to resolve two specific problems associated with the services. One of them is related to burst retransmission in data applications, where a single lost cell in a burst results in retransmission of the entire burst. The other problem is connected with the unpredictable nature of data applications, which results in the difficulty of obtaining accurate traffic characteristics of the connections in advance.

The former problem is associated with UPC (see Section 1.5.3 and Chapter 4) and BMS mechanisms (see Section 1.5.4 and Chapter 5). In these mechanisms, cells are discarded without reference to the bursts they belong to. This means that the lost cells may be spread over several bursts instead of being concentrated in a single burst, and hence the need for retransmitting a large number of bursts leading to large cell losses. We refer to this effect as *cell loss multiplication effect*. This effect results in a dramatic decrease in the effective throughput. In order to minimise the number of retransmitted bursts, a very low cell loss probability is required if cell-level control is used [Turner, 1992]. An alternative to this is to use burst-level control, such as a burst-level policing scheme, proposed by Bala *et al.* [1990], where cell marking is done on a burst basis, and a burst multiplexing scheme, investigated by Callegati and Widjaja [1993], where the multiplexer discards the remainder of a burst if one cell of the burst fails to access the multiplexer's buffer. These burst-level controls, however, do not address the problem of obtaining accurate traffic characteristics of the connections in advance.

The latter problem, concerning the difficulty of obtaining accurate traffic of a connection in advance, is associated with CAC mechanisms. Without accurate traffic characteristics the approches described in Section 1.5.2 and Chapter 3 may allocate too much, or too little, bandwidth to a data connection. A possible solution to the problem is to adapt the bandwidth allocation on the burst-level. In [Rubin and Lin, 1993; Haas and Winters, 1991], the negotiated traffic parameters, and thus the parameters of the input flow-control or policing mechanism, are adapted based on the network loading. The feasibility of such approaches is limited by the propagation delay, as with a long propagation delay the network loading may have changed by the time the source received the control message from the network. On the other hand, Ramamurthy and Dinghe [1991] suggested an adaptation of the negotiated traffic parameters when a burst is generated.

An alternative to adapting the negotiated traffic parameters is to use burst admission control [Boyer, 1990]. This allows a bursty traffic source to notify the network of an impending burst of information and to request rapid allocation of network resources in terms of link bandwidth and buffer spaces in order to ensure delivery of the information. The bandwidth reservation protocol is described as *fast reservation protocol (FRP)* in [Boyer and Tranchier, 1992], while the buffer reservation protocol is described as fast buffer reservation in [Turner, 1992] and burst intelligent multiplexing in [Dutkiewitz *et al.*, 1992].

Recently *FRP with delayed retransmission (FRP/DT)*, described by Boyer and Tranchier [1992], has gained popularity since it offers a solution to both problems. In FRP/DT bursts are sent only when the request for network resources is accepted, which implies a lossless transmission. Performance studies of FRP/DT indicate that the protocol is good for long bursts and short round trip delays [Boyer and Tranchier, 1992]. However, with short bursts and long round trip delays, FRP/DT can lead to low resource utilisation since the sources can not transmit before confirmation of reservation for resources is received; during this round trip delay, the reserved resources can not be utilised by other sources and hence are wasted. Furthermore the end-to-end cell transfer delay may be intolerable for some services, such as interactive data services. In order to overcome these drawbacks, a modified version of FRP/DT, called *FRP with intermediate transmission (FRP/IT)*, has been proposed [Boyer and Tranchier, 1992]. FRP/IT allows sources to transmit bursts immediately after a reservation is sent without waiting for the acknowledgement. The bursts, however, may be discarded if the reservation fails due to lack of resources. The need to retransmit these bursts will result in the same level of delays as under FRP/DT. This means that the FRP/IT only solves the problem with short bursts but no with long round trip delays.

It was suggested in [Boyer and Tranchier, 1992] to have FRP/DT and FRP/IT coexist in the network to provide an unified control for data applications. However, the use of FRP/IT can not resolve the inefficiency in FRP/DT, which can be illustrated as follows. Say FRP/DT has reserved resources for source A, but the source has not started transmission because of the propagation delay of the acknowledgement message back to the source. Assume that during this period, a reservation is made by source B using FRP/IT and following the reservation cell, source B transmits several bursts. If no resources are available, then the incoming bursts will be lost, even though the resources reserved for source A unused.

In order to overcome the drawback of FRP/DT, we propose that cell-level control, instead of FRP/IT, coexists with FRP/DT. In other words, we propose a combination of cell-level and burst-level control. Such a scheme is currently under investigation and initial results on its performance will be presented in [Hartanto *et al.*, 1994b]. The scheme comprises a combined cell-level and burst-level multiplexer (CCBM) and a combined cell-level and burst-level policer (CCBP), as discussed in the following sections.

Combined Cell-Level and Burst-Level Multiplexer (CCBM)

A *combined cell-level and burst-level multiplexer (CCBM)* is shown in Figure 6.3. In this scheme, instead of utilising the entire link bandwidth for either cell-level multiplexing, as in ATM statistical multiplexing, or burst-level multiplexing, as under FRP, the link bandwidth θ is divided into two parts: one for the cell transmission mode θ_c, and the other for the burst transmission mode θ_b, which can cater for $N_b = \theta_b/\lambda_b$ bursts, where λ_b is the peak rate of the bursts. Arriving

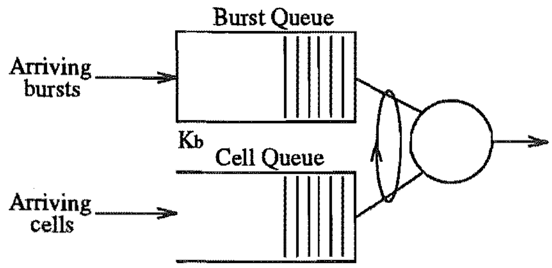


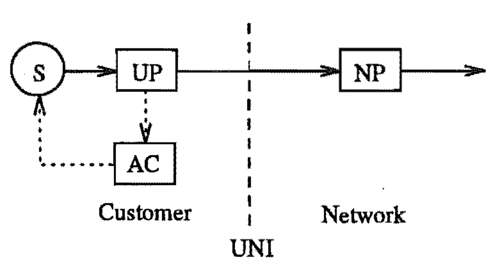
Figure 6.3 The combined cell-level and burst-level multiplexer (CCBM).

bursts and arriving cells are queued separately and we assume that the buffer for Cell Queue is chosen large enough to avoid any loss of cells.

The scheme is similar to DQCS, proposed in Chapter 5, except that the Burst Queue utilises burst multiplexing, instead of cell multiplexing. The queues are served according to a modified exhaustive limited discipline. If a service cycle M slots, then the server will serve the Burst Queue up to $M_b = M\theta_b^*/\theta$ slots, where $\theta_b^* = N_b^*\lambda_b$ and N_b^* is the number of active bursts ($N_b^* \leq N_b$). The server will then switch to serve the Cell Queue for a maximum of $M_c = M - M_p$ slots. Such a service implies that any unused bandwidth from the Burst Queue can be utilised by the Cell Queue, but the converse is not true, because we want to ensure that the bursts are transmitted at λ_b. Such sharing of bandwidth by the Cell Queue means that this queue can use the idle bandwidth after a reservation has been made, but before the source has received the acknowledgement and has started transmitting bursts. This allows the queue to be depleted faster thereby reducing the cell delays at the queue.

Combined Cell-Level and Burst-Level Policer (CCBP)

As with other services, a policer is required to ensure that the traffic generated by users conforms to the negotiated traffic parameters. In order to avoid any cells being discarded by the policer at the access nodes, it is preferable for users to police their traffic properly and to buffer any excess traffic, while the network needs only to verify the policing. A general framework for such a policer, called *user-network policer (UNP)*, has previously been proposed by the author [Hartanto and Sirisena, 1993a]. The functional scheme for the UNP is shown in Figure 6.4



S = Source; AC = Adaptive control
UP = User policer; NP = Network policer

Figure 6.4 User-network policing (UNP).

The *user policer (UP)* belongs to the flow control functions operated at the user terminals, where users have the flexibility to select the most appropriate scheme to suit their applications. On the other hand, the network can implement a *network policer (NP)* to match the class of connection, either pure or mixed. The UNP offers a more general framework than shaper-enforcer [Woodruff *et al.*, 1988] or flow throttling-flow enforcement [Sallberg *et al.*, 1990], as it allows users to police cells or bursts and to selectively mark them. It also allows users to implement additional *adaptive control (AC)* schemes. For example, in policing a data connection, a user-policer which can police both the average rate and the peak rate of the source may be as shown in Figure 6.5

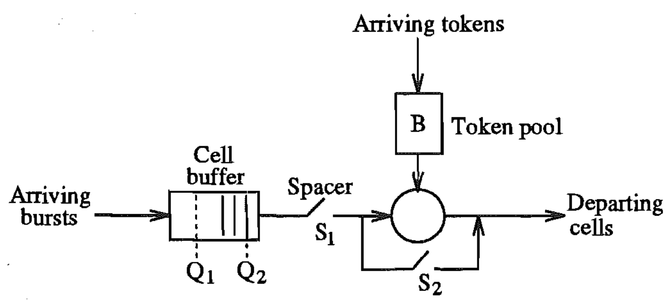


Figure 6.5 Combined cell-level and burst-level policer (CCBP).

The proposed policing scheme, referred to as *combined cell-level and burst-level policer (CCBP)*, consists of a spacer and a leaky bucket. The spacer can be viewed as a switch S_1, which closes only after a spacing period $T_b(T_b = 1/\lambda_b)$ has elapsed since the last cell departure, otherwise S_1 is open. In the cell transmission mode, the leaky bucket is used to shape the average rate of the traffic, while the spacer is inserted to limit the peak rate. However, when the source switches to burst transmission mode, then the leaky bucket can be bypassed by closing switch S_2, resulting in cells being spaced out only without being policed. Such bypassing of the leaky bucket reduces the need for modifying the parameters of the leaky bucket. The peak and the average rates can be negotiated during the connection setup. If necessary, users can still renegotiate the parameters during the connection holding time, but this will not be carried out for every burst as under FRP or in [Ramamurthy and Dinghe, 1991].

The buffer thresholds Q_1 and Q_2 facilitate an adaptive strategy which involve a switching from cell transmission mode to burst transmission mode, and vice versa. The threshold Q_1 is used for indicating a switch to burst transmission mode. If the queue length is larger than Q_1 then the user will send a reservation cell to reserve the required peak bandwidth. On the other hand, if the queue length drops below Q_2 or the queue becomes empty, then another cell will be sent to the network in order to release the peak bandwidth and switch to the cell transmission mode. Users can choose the buffer thresholds according to the burst lengths of their applications. Users also have the freedom to choose either to keep transmitting after sending a reservation cell (similar to FRP/IT, except no cell will be lost), or to stop transmitting and to wait until an acknowledgement has been received (similar to FRP/DT). In the latter case, token generation is suspended when the reservation for the peak rate has been sent to avoid accumulation of tokens during burst transmission mode which would cause a sudden burst into the network when switching to cell transmission mode. The token generation is resumed when the CCBP switches back to the cell transmission mode.

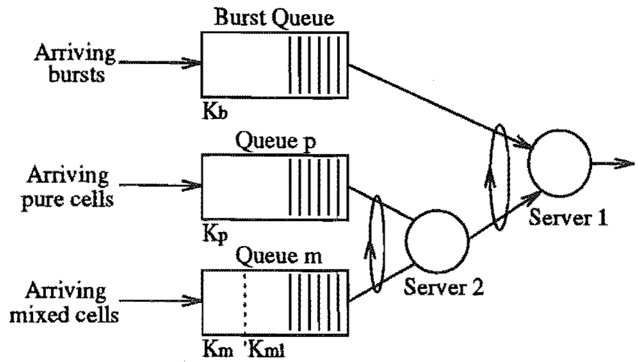


Figure 6.6 Hierarchical dual queues with cyclic service (HDQCS).

A possible integration of the scheme with voice and video applications can be achieved by using a *hierarchical dual queue with cyclic service (HDQCS)* as shown in Figure 6.6. In the scheme, the allocated bandwidth θ_c for Cell Queue is further partitioned into θ_{cp} for Queue p and θ_{cm} for Queue m. If Server 1 is in the cell transmission mode, then it will trigger Server 2 to serve either Queue p or Queue m using an exhaustive limited discipline. This queue bears some resemblance to the hierarchical weighted round robin (HWRR), which has been used in a delay priority scheme [Hluchyj and Bhargava, 1992].

6.2.3 Overall Traffic Management Framework

In addition to offering a unified control for data applications, CCBM, CCBP and HDQCS, described in the previous section, potentially offer also a full integration of the traffic mangement framework for B-ISDNs. Together with the other control schemes discussed in Chapters 3-5, we could propose an overall traffic management for B-ISDNs as shown in Figure 6.7. The italic lettering indicates the methods investigated in this thesis which are suggested for the required control, e.g. BLBP scheme is suggested for policing a voice or video traffic of mixed classs.

For CBR voice or video services, where cells are generated periodically and no cells can be tagged, we can allocate a peak bandwidth to the services and treat them as data services using burst-transmission mode with burst size equal to one. For these services, we only need to enforce the peak rate by spacing the cells. On the other hand, for VBR voice or video services generating bursty traffic without pretagged cells, the bandwidth required can be found using Method I (see Chapter 3). These services will make use of cell transmission mode. During cell transmission, a traffic shaper, similar to CCBP, can be used as a user policer, while the network can use LB or BLB. If the shaper buffer builds up, then an adaptive encoder can be employed to reduce the number of bits per sample as in [Yin and Hluchyj, 1991]. For VBR voice or video services which allow pretagging or marking of cells, the bandwidth required can be found using Method V (see Chapter 3). The network can use BLBP or DLBP or BDLBP as the network policer.

For data services, where the ATM networks are required to cater for burst size ranging from tens (e.g. 33 for the Ethernet) to few thousands cells (e.g. for large file transfers), a combination of cell transmission mode and burst transmission mode as proposed in the previous section can be used. The choice of either mode is up to the users. For example, initially users can choose

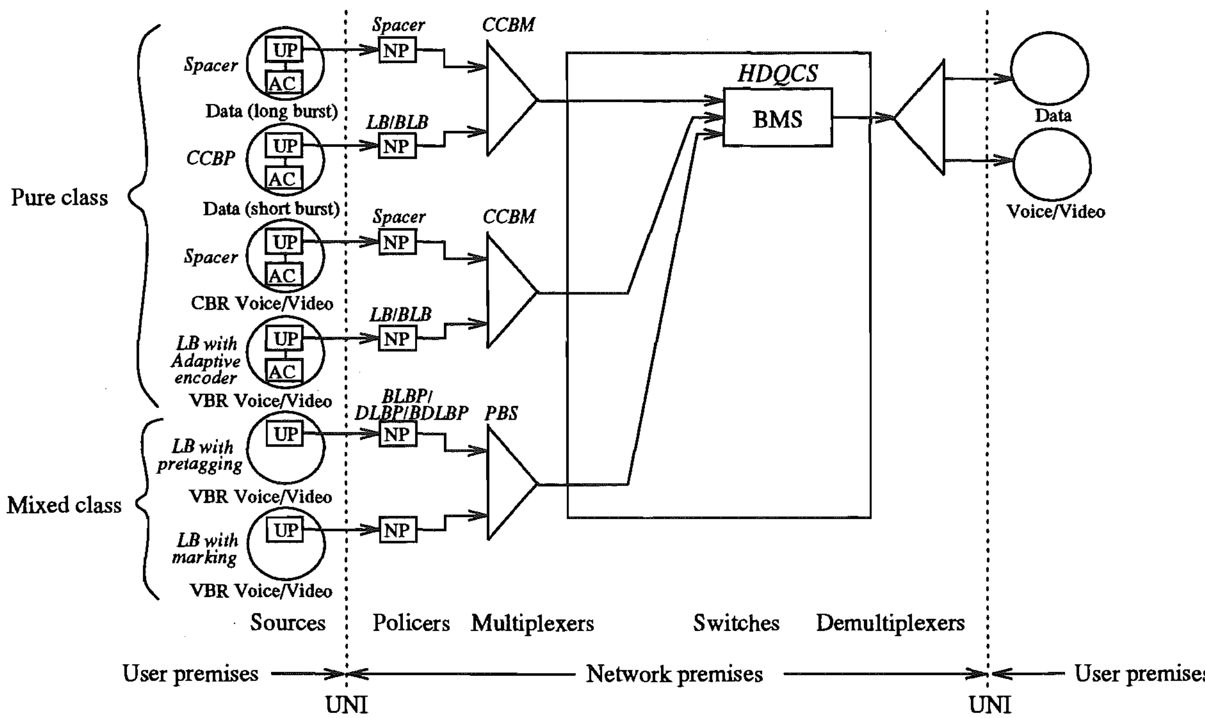


Figure 6.7 Overall traffic management framework.

the cell transmission mode and the network could allocate the resources based on a conservative average rate using Method I. If the users keep transmitting short bursts, then the likelihood of the Cell buffer in CCBP being filled up will be small, hence there would be no need for switching to burst transmission mode. On the other hand, if users keep transmitting long bursts then it is likely that they would have to switch to burst transmission mode for every burst generated. This latter will be equivalent to using FRP/DT.

The overall traffic management framework, proposed in this section, allows users to choose which control techniques to be used depending on the required applications. We expect that such a user-network oriented approach can provide a more unified control scheme within the network than one based on the network-oriented approach only. Further research is needed to evaluate and fine tune this proposed overall traffic management framework.

REFERENCES

ADDIE, R. AND ZUKERMAN, M. (1993a), 'An Approximation for Performance Evaluation of Stationary Single Server Queues', In *Proc. of IEEE INFOCOM Conference*, San Francisco, CA, March, pp. 835–842.

ADDIE, R. AND ZUKERMAN, M. (1993b), 'Analysis of Connection Admission Control and Multiplexing Gain in a B-ISDN based on a Gaussian Traffic Model', In *Proc. of Australian Broadband Switching and Services Symposium*, Wollongong, Australia, July, pp. 1–8.

AHMADI, H. AND DENZEL, W. (1989), 'A Survey of Modern High Performance Switching Technologies', *IEEE Journal on Selected Areas in Communications*, Vol. 7, No. 7, September, pp. 1091–1103.

AKSELSEN, S., EIDSVIK, A. AND FOLKOW, T. (1993), 'Telemedicine and ISDN', *IEEE Communication Magazine*, Vol. 31, No. 1, January, pp. 46–51.

AMIN-SALEHI, B., FLINCHBAUGH, G. AND PATE, L. (1990), 'Implications of New Network Services on B-ISDN Capabilities', In *Proc. of IEEE INFOCOM Conference*, San Francisco, CA, June, pp. 1038–1046.

ANGGAWIJAYA, H. (1994), *Congestion Control in ATM*, Master's thesis, University of Canterbury, New Zealand.

ANICK, S., MITRA, D. AND SONDHI, M. (1982), 'Stochastic Theory of A Data-Handling Systems with Multiple Sources', *Bell System Technical Journal*, Vol. 61, No. 8, October, pp. 1871–1894.

ANIDO, G. (1989), 'Traffic Control and Management Mechanisms for a Broadband Packet Network', In *Proc. of ITC Specialist Seminar*, Adelaide, Australia, September.

ARMBRUSTER, H. AND SCHAFFER, B. (1986), 'Broadband ISDN: Realization Aspects of the Future', *Telecom Report International*, Vol. 10, No. 5, pp. 206–216.

AWATER, G. AND SCHOUTE, F. (1991), 'Optimal Queueing Policies for Fast Packet Switching of Mixed Traffic', *IEEE Journal on Selected Areas in Communications*, Vol. 9, No. 3, April, pp. 458–467.

BAE, J. AND SUDA, T. (1991), 'Survey of Traffic Control Schemes and Protocols in ATM Network', *Proceedings of IEEE*, Vol. 79, No. 2, February, pp. 170–189.

BAIOCCHI, A., BLEFARI-MELLAZZI, N., LISTANI, M., ROVERI, A. AND WINKLER, R. (1991), 'Load Performance Analysis of An ATM Multiplexer Loaded with High-Speed ON-OFF Sources', *IEEE Journal on Selected Areas in Communications*, Vol. 9, No. 3, April, pp. 388–393.

BAIOCCHI, A., BLEFARI-MELLAZZI, N. AND SALVATORE, F. (1992), 'On the Significant Parameters for the Characterization of the Cell Loss Behaviour in ATM Multiplexing', In *Proc. of IEEE ICC Conference*, Chicago, MI, June, pp. 682–693.

BALA, K., CIDON, I. AND SOHRABY, K. (1990), 'Congestion Control for High Speed Packet Switched Networks', In *Proc. of IEEE INFOCOM Conference*, San Francisco, CA, June, pp. 520–526.

BALLART, R. (1989), 'SONET - Now It's the Standard Optical Networks', *IEEE Communication Magazine*, Vol. 29, No. 3, March, pp. 8–15.

BERALDI, R. AND MARANAO, S. (1992), 'Limiting Removal Depth in the Pushout Scheme for ATM Networks', In *Proc. of IEEE ICC Conference*, Chicago, MI, June, pp. 997–1003.

BERGER, A. (1991), 'Performance Analysis of A Rate-Control Throttle where Tokens and Jobs Queue', *IEEE Journal on Selected Areas in Communications*, Vol. 9, No. 2, February, pp. 165–170.

BONOMI, F., FRATTA, L., MONTAGNA, S. AND PAGLINO, R. (1990), 'Priority on Cell Service and on Cell Loss in ATM Switching', In *Proc. of ITC Specialist Seminar*, Morristown, NJ, October.

BONOMI, F., MONTAGNA, S. AND PAGLINO, R. (1993), 'A Further Look at Statistical Multiplexing in ATM Networks', *Computer Networks and ISDN Systems*, Vol. 26, No. 1, September, pp. 119–138.

BOWERS, J. AND BENFORD, S. (EDS.) (1991), *Studies in Computer Supported Collaborative Work: Theory, Practice and Design*, Elsevier Science Publishers, B.V., North-Holland.

BOYER, P. (1990), 'A Congestion Control for the ATM', In *Proc. of ITC Specialist Seminar*, Morristown, NJ, October.

BOYER, P. AND TRANCHIER, D. (1992), 'A Reservation Principle with Applications to the ATM Traffic Control', *Computer Networks and ISDN Systems*, Vol. 24, No. 4, May, pp. 321–334.

BOYER, P., GUILLEMIN, F., SERVEL, M. AND COUDREUSE, J. (1992), 'Spacing Cells Protects and Enhances Utilization of ATM Network Links', *IEEE Network Magazine*, Vol. 6, No. 5, September, pp. 38–49.

BUBENIK, R., DEHART, J. AND GADDIS, M. (1991), 'Multipoint Connection Management in High Speed Networks', In *Proc. of IEEE ICC Conference*, Denver, CO, June, pp. 59–68.

BURGIN, J. (1989), 'Virtual Path Management in B-ISDN', In *Proc. of the Fourth Australian Fast Packet Switching Workshop*, Sydney, Australia, July, pp. 1–16.

BURGIN, J. AND DORMAN, D. (1991), 'Broadband ISDN Resource Management: The Role of Virtual Paths', *IEEE Communication Magazine*, Vol. 29, No. 9, September, pp. 44–48.

BUTTO, M., CAVALLERO, E. AND TONIETTI, A. (1991), 'Effectiveness of the Leaky Bucket Policing Mechanisms in ATM Network', *IEEE Journal on Selected Areas in Communications*, Vol. 9, No. 3, April, pp. 335–342.

CALLEGATI, F. AND WIDJAJA, I. (1993), 'Burst Multiplexing versus Cell Multiplexing', In *Proc. of the Eight Australian Teletraffic Research Seminar*, Melbourne, Australia, December, pp. 66–73.

CCITT (1990a), 'Draft Recommendation I.121: Broadband Aspects of ISDN', SG XVIII, June.

CCITT (1990b), 'Draft Recommendation I.211: B-ISDN Service Aspects', SG XVIII, June.

CCITT (1990c), 'Draft Recommendation I.311: B-ISDN General Network Aspects', SG XVIII, June.

CCITT (1990d), 'Draft Recommendation I.321: B-ISDN Protocol Reference Model and Its Application', SG XVIII, June.

CCITT (1990e), 'Draft Recommendation I.362: B-ISDN ATM Adaptation Layer (AAL) Functional Description', SG XVIII, June.

CCITT (1990f), 'Draft Recommendation I.432: B-ISDN User-Network-Interface - Physical Layer Specification', SG XVIII, June.

CCITT (1992a), 'Draft Recommendation I.361: B-ISDN ATM Layer Specification', SG XVIII, June.

CCITT (1992b), 'Draft Recommendation I.362: B-ISDN ATM Adaptation Layer (AAL) Functional Description', SG XVIII, June.

CCITT (1992c), 'ISCP Baseline Document (Version 3.0)', SG XI, November.

CHANG, W., CHANG, N. AND MESSERSCHMITT, D. (1992), 'Call Processing and Signalling in a Desktop Multimedia Conferencing System', In *Proc. of IEEE GLOBECOM Conference*, Orlando, FL, December, pp. 225–229.

CHAO, H. (1991), 'Design of Leaky Bucket Access Control Schemes in ATM Networks', In *Proc. of IEEE ICC Conference*, Denver, CO, June, pp. 180–187.

CHEN, X. AND AKYILDIZ, I. (1992), 'A Finite Buffer Two Class Queue with Different Scheduling and Push-Out Schemes', In *Proc. of IEEE INFOCOM Conference*, Florence, Italy, May, pp. 231–241.

CHEN, X. AND HAYES, J. (1992), 'Call Scheduling in Multicasting Packet Switching', In *Proc. of IEEE ICC Conference*, Chicago, MI, June, pp. 895–899.

CHEN, D. AND MARK, J. (1992), 'Delay and Loss Control of an Output Buffered Fast Packet Switch Supporting Integrated Services', In *Proc. of IEEE ICC Conference*, Chicago, MI, June, pp. 985–989.

CHEN, T., WARLAND, J. AND MESSERSCHMITT, D. (1989), 'Dynamic Priority Protocols for Packet Voice', *IEEE Journal on Selected Areas in Communications*, Vol. 7, No. 5, June, pp. 632–643.

CHUAH, M. AND CRUZ, R. (1990), 'Approximate Analysis of Average Performance of (σ, ρ) Regulators', In *Proc. of IEEE INFOCOM Conference*, San Francisco, CA, June, pp. 874–880.

CIDON, I., GUERIN, R. AND KHAMISY, A. (1993), 'On Protective Buffer Policies', In *Proc. of IEEE INFOCOM Conference*, San Francisco, CA, March, pp. 1051–1058.

COOLEGEM, K., CLARK, W. AND CERUTI, R. (1991), 'Multimedia Desktop Conferencing with MIAS', In *Telecommunication Access Networks Technology and Service Trends - W. Lemstra (ed.)*, Elsevier Science Publishers, B.V., North-Holland, pp. 36–42.

CZACHORSKI, T., FOURNEAU, J. AND PEKERGIN, F. (1992), 'Diffusion Model of the Push-Out Buffer Management Policy', In *Proc. of IEEE INFOCOM Conference*, Florence, Italy, May, pp. 252–261.

DAIGLE, J. AND LANGFORD, J. (1986), 'Models for Analysis of Packet Voice Communications Systems', *IEEE Journal on Selected Areas in Communications*, Vol. 4, No. 6, September, pp. 847–855.

DAY, A. (1991), 'International Standardization of BISDN', *IEEE Lightwave Communication Systems Magazine*, Vol. 2, No. 3, August, pp. 13–20.

DECINA, M. AND TONIATTI, T. (1990), 'On Bandwidth Allocation to Bursty Virtual Connections in ATM Networks', In *Proc. of IEEE ICC Conference*, Atlanta, GA, April, pp. 844–851.

DE PRYCKER, M. (1991), *Asynchronous Transfer Mode: Solution for Broadband ISDN*, Ellis Horwood Ltd., England.

DITTMAN, L., JACOBSEN, S. AND MOTH, K. (1991), 'Flow Enforcement Algorithms for ATM Networks', *IEEE Journal on Selected Areas in Communications*, Vol. 9, No. 3, April, pp. 343–350.

DOSHI, B., ECKBERG, A., SAKSENA, V. AND ZOCCOLILLO, R. (1990), 'A B-ISDN/ATM Network Congestion Control Architecture, & Its Performance with Complementary End-Terminal Flow & Error Controls', In *Proc. of ITC Specialist Seminar*, Morristown, NJ, October.

DURAN, J. AND VISSER, J. (1992), 'International Standards for Intelligent Networks', *IEEE Communication Magazine*, Vol. 31, No. 2, February, pp. 34–42.

DUTKIEWICZ, E. AND ANIDO, G. (1990), 'Connection Admission Control in ATM Networks', In *Proc. of the Fifth Australian Teletraffic Research Seminar*, Melbourne, Australia, December.

DUTKIEWITZ, E., ANIDO, G. AND FOLLETT, D. (1992), 'Fast Resource Management Architecture for International Networks', In *Proc. of Australian Broadband Switching and Services Symposium*, Melbourne, Australia, July, pp. 169–175.

DZIONG, Z. AND CHOQUETTE, J. (1990), 'Admission Control and Routing in ATM Networks', *Computer Networks and ISDN Systems*, Vol. 20, No. 1-5, pp. 189–196.

ECKBERG, A. (1992), 'B-ISDN/ATM Traffic and Congestion Control', *IEEE Network Magazine*, Vol. 6, No. 5, September, pp. 28–37.

ECKBERG, A., LUAN, D. AND LUCANTONI, D. (1989), 'Meeting the Challenge: Congestion and Flow Control Strategies for Broadband Information Transfer', In *Proc. of IEEE GLOBECOM Conference*, Dallas, TX, November, pp. 1769–1773.

EIGEN, D. (1990), 'Narrowband and Broadband ISDN CPE Directions', *IEEE Communication Magazine*, Vol. 28, No. 4, April, pp. 39–46.

ELWALID, A. AND MITRA, D. (1991), 'Analysis and Design of Rate Based Congestion Control of High Speed Networks, I: Stochastic Fluid Models, Access Regulation', *Queueing System*, Vol. 9, No. 1-2, October, pp. 29–64.

ELWALID, A. AND MITRA, D. (1992), 'Fluid Models for the Analysis and Design of Statistical Multiplexing with Loss Priorities on Multiple Classes of Bursty Traffic', In *Proc. of IEEE INFOCOM Conference*, Florence, Italy, May, pp. 415–425.

ENG, K. AND KAROL, M. (1991), 'The Growable Switch Architecture: A Self-Routing Implementation for Large ATM Application', In *Proc. of IEEE ICC Conference*, Denver, CO, June, pp. 1014–1020.

ESAKI, H., IWAMURA, K. AND KODAMA, R. (1990), 'A Simple and Effective Admission Control Method for an ATM Network', In *Proc. of IEEE GLOBECOM Conference*, San Diego, CA, December, pp. 28–33.

ESCOBAR, J., DEUTSCH, D. AND PARTRIDGE, C. (1992), 'Flow Synchronization Protocol', In *Proc. of IEEE GLOBECOM Conference*, Orlando, FL, December, pp. 1381–1387.

FAN, Y. (1991), 'New Policy for Multiqueue Systems with Finite Queue Size', *IEE Elect. Lett.*, Vol. 27, No. 16, August, pp. 1434–1435.

FELLER, W. (1950), *An Introduction to Probability Theory and Its Applications*, John Wiley and Sons, Inc., New York, NY.

FERRARI, D. (1990), 'Client Requirements for Real-Time Communication Services', *IEEE Communication Magazine*, Vol. 28, No. 11, November, pp. 65–72.

FILIPIAK, J. (1989), 'M-Architecture: A Structural Model of Traffic Management and Control in Broadband ISDNs', *IEEE Communication Magazine*, Vol. 27, No. 5, May, pp. 25–31.

FOWLER, H. AND LELAND, W. (1991), 'Local Area Network Traffic Characteristics, with Implications for Broadband Network Congestion Management', *IEEE Journal on Selected Areas in Communications*, Vol. 9, No. 7, September, pp. 1139–1149.

FRIESEN, V. AND WONG, J. (1993), 'The Effect of Multiplexing, Switching and Other Factors on the Performance of Broadband Networks', In *Proc. of IEEE INFOCOM Conference*, San Francisco, CA, March, pp. 1194–1203.

FROST, V. AND MELAMED, B. (1994), 'Traffic Modeling for Telecommunications Networks', *IEEE Communication Magazine*, Vol. 32, No. 3, March, pp. 70–81.

GADDIS, M., BUBENIK, R. AND DEHART, J. (1992), 'A Call Model for Multipoint Communication in Switched Networks', In *Proc. of IEEE ICC Conference*, Chicago, MI, June, pp. 609–615.

GALLASSI, G., RIGOLIO, G. AND FRATTA, L. (1989), 'ATM: Bandwidth Assignment and Bandwidth Enforcement Policies', In *Proc. of IEEE GLOBECOM Conference*, Dallas, TX, November, pp. 1788–1793.

GALLASSI, G., RIGOLIO, G. AND VERRI, L. (1990a), 'Resource Management and Dimensioning in ATM Networks', *IEEE Network Magazine*, Vol. 4, No. 3, May, pp. 8–17.

GALLASSI, G., RIGOLIO, G. AND FRATTA, L. (1990b), 'Bandwidth Assignment in Prioritized ATM Networks', In *Proc. of IEEE GLOBECOM Conference*, San Diego, CA, December, pp. 852–856.

GARCIA, J. AND CASALS, O. (1992), 'Stochastic Models of Space Priority Mechanisms with Markovian Arrival Processes', *Annals of Operations Research*, Vol. 35, April, pp. 271–296.

GARRETT, M. AND WILLINGER, W. (1994), 'Analysis, Modeling and Generation of Self-Similar VBR Video Traffic', In *Proc. of ACM SIGCOMM Conference*, London, U.K., September.

GERLA, M. AND KLEINROCK, L. (1980), 'Flow Control: A Comparative Survey', *IEEE Transaction on Communications*, Vol. 28, No. 4, April, pp. 553–573.

GOLESTANI, S. (1991), 'A Framing Strategy for Congestion Management', *IEEE Journal on Selected Areas in Communications*, Vol. 9, No. 7, September, pp. 1064–1077.

GRAVEY, A., BOYER, P. AND HEBUTERNE, G. (1991), 'Tagging versus Strict Rate Enforcement in ATM Networks', In *Proc. of IEEE GLOBECOM Conference*, Phoenix, AZ, December, pp. 271–275.

GUERIN, R., AHMADI, H. AND NAGHSHINEH, M. (1991), 'Equivalent Capacity and Its Application to Bandwidth Allocation in High-Speed Networks', *IEEE Journal on Selected Areas in Communications*, Vol. 9, No. 7, September, pp. 968–981.

GUILLEMIN, F. AND MONIN, W. (1992), 'Management of Cell Delay Variation in ATM Networks', In *Proc. of IEEE GLOBECOM Conference*, Orlando, FL, December, pp. 128–132.

HAAS, Z. AND WINTERS, J. (1991), 'Congestion Control for Adaptive Admission', In *Proc. of IEEE INFOCOM Conference*, Bal Harbour, FL, April, pp. 560–569.

HANDEL, R. (1989), 'Evolution of ISDN Towards Broadband ISDN', *IEEE Network Magazine*, Vol. 3, No. 1, January, pp. 4–13.

HARTANTO, V. (1993), 'DESC++: Discrete Event Simulation Library using C++ Language', *Technical report*, University of Canterbury, New Zealand, January.

HARTANTO, V. AND SIRISENA, H. (1993a), 'User-Network Policer: A New Approach for ATM Congestion Control', In *Proc. of IEEE INFOCOM Conference*, San Francisco, CA, March, pp. 376–383.

HARTANTO, V. AND SIRISENA, H. (1993b), 'A Framework for B-ISDN CPE and Signalling', In *Proc. of IEEE GLOBECOM Conference*, Houston, TX, November, pp. 1511–1515.

HARTANTO, V., SIRISENA, H., PAWLIKOWSKI, K. AND KENNEDY, W. (1991), 'Performance Study of Dual Queues with Limited Cyclic Service in ATM Switching', In *Proc. of Singapore Int. Conf. on Networks*, Singapore, September, pp. 253–258.

HARTANTO, V., SIRISENA, H., PAWLIKOWSKI, K. AND KENNEDY, W. (1992), 'Dissecting Call Establishment Procedures in ATM Networks', In *Proc. of Australian Broadband Switching and Services Symposium*, Melbourne, Australia, July, pp. 605–612.

HARTANTO, V., PAWLIKOWSKI, K., SIRISENA, H. AND KREUTZER, W. (1994a), 'DESC++: An Object-Oriented Tool for Simulating Telecommunication Networks', *submitted for publication in Computers and Electrical Engineering*.

HARTANTO, V., SIRISENA, H. AND PAWLIKOWSKI, K. (1994b), 'Comparative Study of Fast Reservation Protocols and Adaptive Input-Rate Flow Controls', In *Australian Telecommunication Networks and Applications Conference*, Melbourne, Australia, December.

HEBUTERNE, G. AND GRAVEY, A. (1989), 'A Space Priority Queueing Mechanism for Multiplexing ATM Channels', In *Proc. of ITC Specialist Seminar*, Adelaide, Australia, September.

HEFFES, H. (1980), 'A Class of Data Traffic Processes - Covariance Function Characterization and Related Qeueuing Results', *Bell System Technical Journal*, Vol. 59, No. 6, July, pp. 897–929.

HEFFES, H. AND LUCANTONI, D. (1986), 'A Markov Modulated Characterization of Packetized Voiced and Data Traffic and Related Statistical Multiplexer Performance', *IEEE Journal on Selected Areas in Communications*, Vol. 4, No. 6, September, pp. 856–868.

HEHMANN, D. (1990), 'Transport Service for Multimedia Application on Broadband Networks', *Computer Communications*, Vol. 13, No. 4, May, pp. 197–203.

HEIDELBERGER, P. AND WELCH, P. (1981), 'A Spectral Method for Confidence Interval Generation and Run Control in Simulations', *Communications of the ACM*, Vol. 24, No. 4, April, pp. 233–245.

HEMMER, H. AND HUTH, P. (1991), 'Evaluation of Policing Functions in ATM Networks', In *Queueing, Performance and Control in ATM (ITC-13) - J.W. Cohen and C.D. Pack (eds.)*, Elsevier Science Publishers, B.V., North-Holland, pp. 111–116.

HENNESSY, J. AND JOUPPI, N. (1991), 'Computer Technology and Architecture: An Evolving Interaction', *IEEE Computer Magazine*, Vol. 24, No. 9, September, pp. 18–29.

HENRICI, P. (1979), 'Fast Fourier Methods in Computational Complex Analysis', *SIAM Review*, Vol. 21, No. 4, October, pp. 481–527.

HEYMAN, D. (1982), 'An Analysis of the Carrier-Sense Multiple-Access Protocol', *Bell System Technical Journal*, Vol. 61, No. 8, October, pp. 2023–2051.

HEYMAN, D., TABATABAI, A. AND LAKSHMAN, T. (1992), 'Statistical Analysis and Simulation Study of Video Teleconference Traffic in ATM Network', *IEEE Trans. Circuits and Systems for Video Tech.*, Vol. 2, No. 1, March, pp. 49–59.

HLUCHYJ, M. AND BHARGAVA, A. (1992), 'Queueing Disciplines for Integrated Fast Packet Networks', In *Proc. of IEEE ICC Conference*, Chicago, MI, June, pp. 990–996.

HOU, T. AND WONG, H. (1990), 'Queueing Analysis for ATM Switching of Mixed Continuous-Bit-Rate and Bursty Traffic', In *Proc. of IEEE INFOCOM Conference*, San Francisco, CA, June, pp. 660–667.

HUANG, A. AND KNAUER, S. (1984), 'Starlite: A Wideband Digital Switch', In *Proc. of IEEE GLOBECOM Conference*, Atlanta, GA, November, pp. 121–125.

HUANG, T., WU, J. AND WU, J. (1993), 'Priority Management to Improve the QOS in ATM Networks', *IEICE Transaction on Communication*, Vol. E76-B, No. 3, March, pp. 249–257.

HUGHES, D., ANIDO, G. AND BRADLOW, H. (1990a), 'Characterising Leaky Bucket Performance for Small Leaky Bucket Depths', In *Proc. of the Fifth Australian Fast Packet Switching Workshop*, Melbourne, Australia, July.

HUGHES, D., ANIDO, G. AND BRADLOW, H. (1990b), 'Queueing Analysis of Multiplexed Bursty Traffic with Application to ATM Switch Performance', In *Proc. of the Fifth Australian Teletraffic Research Seminar*, Melbourne, Australia, December.

HUGHES, D., ANIDO, G. AND BRADLOW, H. (1992), 'Peak Rate Enforcement in B-ISDN: A Worst Case Analysis', *Australian Telecommunication Review*, Vol. 26, No. 2, pp. 1–10.

HUI, J. (1988), 'Resource Allocation for Broadband Networks', *IEEE Journal on Selected Areas in Communications*, Vol. 6, No. 9, December, pp. 1598–1608.

HUI, J. AND RENNER, T. (1990), 'Queueing Strategies for Multicast Packet Switching', In *Proc. of IEEE GLOBECOM Conference*, San Diego, CA, December, pp. 1431–1437.

IDE, I. (1988), 'Superposition of Interrupted Poisson Processes and Its Application to Packetized Voice Multiplexers', In *Teletraffic Science for New Cost-Effective Systems, Networks and Services - M. Bonatti (ed.)*, Elsevier Science Publishers, B.V., North-Holland, pp. 1399–1405.

ILYAS, M. AND MOUFTAH, H. (1990), 'Performance Evaluation of Congestion Avoidance in Broadband ISDNs', In *Proc. of IEEE ICC Conference*, Atlanta, GA, April, pp. 727–731.

ISDN (1991), 'Special Issue on ISDN', *Proceedings of IEEE*, Vol. 79, No. 2, February.

ISDN (1992), 'ISDN Special Issue: End User Requirements and Solutions', *IEEE Communication Magazine*, Vol. 30, No. 8, August.

JABBARI, B. (1990), 'A Bandwidth Allocation Technique for High Speed Networks', In *Proc. of IEEE GLOBECOM Conference*, San Diego, CA, December, pp. 355–359.

JAIN, R. AND ROUTHIER, S. (1986), 'Packet Trains - Measurements and a New Model for Computer Network Traffic', *IEEE Journal on Selected Areas in Communications*, Vol. 4, No. 6, October, pp. 986–995.

JAISWAL, N. (1968), *Priority Queues*, Academic Press, New York, NY.

JEFFREY, M. AND SARMA, A. (1992), 'A Signalling Architecture for Multimedia Services', In *Proc. of IEEE ICC Conference*, Chicago, MI, June, pp. 604–608.

JENG, F. AND LEE, S. (1991), 'Concealment of Bit Error and Cell Loss in Inter-frame Coded Video Transmission', In *Proc. of IEEE ICC Conference*, Denver, CO, June, pp. 496–499.

JOOS, P. AND VERBIEST, W. (1989), 'A Statistical Bandwidth Allocation and Usage Monitoring Algorithm for ATM Networks', In *Proc. of IEEE ICC Conference*, Boston, MA, June, pp. 416–422.

KALMANEK, C. AND KANAKIA, H. (1990), 'Rate Controlled Servers for Very High-Speed Networks', In *Proc. of IEEE GLOBECOM Conference*, San Diego, CA, December, pp. 12–20.

KAMITAKE, T. AND SUDA, T. (1989), 'Evaluation of an Admission Control Scheme for an ATM Network Considering Fluctuations in Cell Loss Rate', In *Proc. of IEEE GLOBECOM Conference*, Dallas, TX, November, pp. 1774–1780.

KIM, Y. AND MONDESTINO, J. (1992), 'Adaptive Entropy Coded Subband Coding of Images', *IEEE Transaction on Image Processing*, Vol. 1, No. 1, January.

KLEINROCK, L. (1975), *Queueing Systems, Vol. 1: Theory*, John Wiley and Sons, Inc., New York, NY.

KLEINROCK, L. (1976), *Queueing System, Vol. 2: Computer Applications*, John Wiley and Sons, Inc., New York, NY.

KONHEIM, A. (1975), 'An Elementary Solution to the Queueing System GI/G/1', *SIAM Journal of Computing*, Vol. 4, No. 4, December, pp. 540–545.

KOWTHA, S. AND VAMAN, D. (1992), 'A Generalized ATM Traffic Model and Its Application in Bandwidth Allocation', In *Proc. of IEEE ICC Conference*, Chicago, MI, June, pp. 1009–1013.

KRONER, H. (1990), 'Comparative Performance Study of Space Priority Mechanisms for ATM Networks', In *Proc. of IEEE INFOCOM Conference*, San Francisco, CA, June, pp. 1136–1143.

KRONER, H., HEBUTERNE, G., BOYER, P. AND GRAVEY, A. (1991), 'Priority Management in ATM Switching Nodes', *IEEE Journal on Selected Areas in Communications*, Vol. 9, No. 3, April, pp. 418–427.

LANG, M. AND BOSCH, M. (1991), 'Performance Analysis of Finite Capacity Polling Systems with Limited-M Service', In *Queueing, Performance and Control in ATM (ITC-13), J.W. Cohen and C.D. Pack (eds.)*, Elsevier Science Publishers, B.V., North-Holland, pp. 731–735.

LAZAR, A., GIDRON, R. AND TEMPLE, A. (1989), 'A Switching Architecture for Asynchronous Time Sharing', In *Proc. of IEEE GLOBECOM Conference*, Dallas, TX, November, pp. 1166–1172.

LE BOUDEC, J. (1991), 'An Efficient Solution Method for Markov Models of ATM Links with Loss Priorities', *IEEE Journal on Selected Areas in Communications*, Vol. 9, No. 3, April, pp. 408–417.

LE BOUDEC, J. (1992), 'The Asynchronous Transfer Mode: A Tutorial', *Computer Networks and ISDN Systems*, Vol. 24, No. 4, May, pp. 279–309.

LEE, T. (1988a), 'M/G/1/N Queue with Vacation Time and Limited Service Discipline', *Performance Evaluation*, Vol. 9, No. 1, November, pp. 181–190.

LEE, T. (1988b), 'Nonblocking Copy Networks for Multicast Packet Switching', *IEEE Journal on Selected Areas in Communications*, Vol. 6, No. 9, December, pp. 1455–1467.

LEE, J. AND LEE, B. (1992), 'Performance Analysis of ATM Cell Multiplexer with MMPP Input', *IEICE Transaction on Communication*, Vol. E75-B, No. 8, August, pp. 709–714.

LELAND, W., TAQQU, M., WILLINGER, W. AND WILSON, D. (1993), 'On the Self-Similar Nature of Ethernet Traffic', In *Proc. of ACM SIGCOMM Conference*, San Francisco, CA, September, pp. 183–193.

LEUNG, W., BAUMGARTNER, T., HWANG, Y., MORGAN, M. AND TU, S. (1990), 'A Software Architecture for Workstations Supporting Multimedia Conferencing in Packet Switching Networks', *IEEE Journal on Selected Areas in Communications*, Vol. 8, No. 3, April, pp. 380–390.

LI, S. (1988), 'Study of Packet Loss in a Packet Switched Voice System', In *Proc. of IEEE ICC Conference*, Philadelphia, PA, June, pp. 1519–1526.

LI, S. (1991), 'A General Solution Technique for Discrete Queueing Analysis of Multimedia Traffic on ATM', *IEEE Transaction on Communications*, Vol. 39, No. 7, July, pp. 1115–1132.

LI, L., KARMOUCH, A. AND GEORGANAS, N. (1992), 'Synchronization in Real Time Multimedia Data Delivery', In *Proc. of IEEE ICC Conference*, Chicago, MI, June, pp. 587–591.

LIAO, K. AND MASON, L. (1989), 'A Discrete-Time Single Server Queue with a Two-Level Modulate', In *Proc. of IEEE GLOBECOM Conference*, Dallas, TX, November, pp. 913–918.

LIM, H. AND SCOTT, C. (1991), 'The Use of Call and Connection Separation to Support a B-ISDN Multiparty and Multimedia Conference Service', In *Proc. of Australian Broadband Switching and Services Symposium*, Sydney, Australia, July, pp. 182–189.

LINDISKY, W. (1990), 'Data Communications Needs', *IEEE Network Magazine*, Vol. 4, No. 2, March, pp. 28–33.

LITTLE, T. AND GHAFOOR, A. (1991a), 'Spatio-Temporal Composition of Distributed Multimedia Objects Value-Added Networks', *IEEE Computer Magazine*, Vol. 24, No. 10, October, pp. 42–50.

LITTLE, T. AND GHAFOOR, A. (1991b), 'Multimedia Synchronization Protocols for Broadband Integrated Services', *IEEE Journal on Selected Areas in Communications*, Vol. 9, No. 9, December, pp. 1368–1382.

LOUVION, J., BOYER, P. AND GRAVEY, A. (1988), 'A Discrete-Time Single Server Queue with Bernoulli Arrivals and Constant Service Time', In *Proc. of the 12th Int. Teletraffic Congress*, Torino, Italy, January.

LY, K., LIM, H. AND SCOTT, C. (1991), 'Design and Implementation Issues for a B-ISDN Multipoint Multi-Media Conference Service', In *Proc. of Australian Broadband Switching and Services Symposium*, Sydney, Australia, July, pp. 33–40.

MAENO, K., SAKATA, S. AND OHMORI, T. (1991), 'Distributed Desktop Conferencing System (MERMAID) Based on Group Communication Architecture', In *Proc. of IEEE ICC Conference*, Denver, CO, June, pp. 520–525.

MAGLARIS, B., ANASTASSIOU, D., SEN, P., KARLSSON, G. AND ROBBINS, H. (1988), 'Performance Models of Statistical Multiplexing in Packet Video Communications', *IEEE Transaction on Communications*, Vol. 36, No. 6, July, pp. 834–844.

MAKRUCKI, B. (1991), 'On The Performance of Submitting Excess Traffic to ATM Networks', In *Proc. of IEEE GLOBECOM Conference*, Phoenix, AZ, December, pp. 281–288.

MASE, K. AND SHODA, S. (1991), 'Real-Time Network Management for ATM Networks', In *Queueing, Performance and Control in ATM (ITC-13), J.W. Cohen and C.D. Pack (eds.)*, Elsevier Science Publishers, B.V., North-Holland, pp. 129–136.

MCKINNEY, R. AND GORDON, T. (1994), 'ATM for Narrowband Services', *IEEE Communication Magazine*, Vol. 32, No. 4, April, pp. 64—72.

MCLAREN, D. (1992), 'Access Control of ATM Packet Video: A Question of Priorities', In *Proc. of Australian Broadband Switching and Services Symposium*, Melbourne, Australia, July, pp. 174–181.

MCLAREN, D. AND NGUYEN, D. (1992), 'Variable Bitrate Source Modelling of ATM-Based Video Services', *Image Communication*, Vol. 4, No. 3, June.

MCNINCH, B. (1990), 'Screen-Based Technology', *IEEE Communication Magazine*, Vol. 28, No. 4, April, pp. 34–38.

MEYER, J., MONTAGNA, S. AND PAGLINO, R. (1993), 'Dimensioning of an ATM Switch with Shared Buffer and Threshold Priority', *Computer Networks and ISDN Systems*, Vol. 26, No. 1, September, pp. 95–108.

MINZER, S. (1989), 'Broadband ISDN and Asynchronous Transfer Mode (ATM)', *IEEE Communication Magazine*, Vol. 27, No. 9, September, pp. 17–24, 57.

MINZER, S. (1991), 'A Signaling Protocol for Complex Multimedia Services', *IEEE Journal on Selected Areas in Communications*, Vol. 9, No. 9, December, pp. 1383–1394.

MITROU, N. AND PENDRAKIS, D. (1991), 'Cell-Level Statistical Multiplexing in ATM Networks: Analysis, Dimensioning and Call-Acceptance Control w.r.t. QOS Criteria', In *Queueing, Performance and Control in ATM (ITC-13), J.W. Cohen and C.D. Pack (eds.)*, Elsevier Science Publishers, B.V., North-Holland, pp. 7–12.

MIYAO, Y. (1991), 'A Call Admission Control Scheme in ATM Networks', In *Proc. of IEEE ICC Conference*, Denver, CO, June, pp. 391–396.

MONTEIRO, J., GERLA, M. AND FRATTA, L. (1991), 'Statistical Multiplexing in ATM Networks', *Performance Evaluation*, Vol. 12, No. 3, pp. 157–167.

MURATA, M., OIE, Y., SUDA, T. AND MIYAHARA, H. (1990), 'Analysis of a Discrete-Time Single-Server Queue with Bursty Inputs for Traffic Control in ATM Networks', *IEEE Journal on Selected Areas in Communications*, Vol. 8, No. 3, April, pp. 447–458.

NAGARAJAN, R., KUROSE, J. AND TOWSLEY, D. (1991), 'Approximation Techniques for Computing Packet Loss in Finite-Buffered Voice Multiplexers', *IEEE Journal on Selected Areas in Communications*, Vol. 9, No. 3, April, pp. 368–377.

NAISH, G. (1994), 'Protocols for Real Time Multimedia Synchronization and Clock Synchronization', *Technical report*, University of Canterbury, New Zealand, January.

NAKAMURA, K., TAKAHASHI, M., HAMADA, N., INUZUKA, T., YANAI, N., MARUO, S. AND KANEDA, K. (1990), 'Personal Multimedia Teleconferencing Terminal', In *Proc. of IEEE ICC Conference*, Atlanta, GA, April, pp. 123–127.

NEUTS, M. (1981), *Matrix-geometric Solutions in Stochastic Models, An Algorithmic Approach*, John Hopkins Univ. Press, Baltimore, MD.

NEWMAN, P. (1992), 'ATM Technology for Corporate Networks', *IEEE Communication Magazine*, Vol. 30, No. 4, April, pp. 90–101.

NEWMAN, P. (1993), 'Backward Explicit Congestion Notification for ATM Local Area Networks', In *Proc. of IEEE GLOBECOM Conference*, Houston, TX, November, pp. 719–723.

NEWMAN, P. AND DOAR, M. (1990), 'A Slotted Ring Copy Fabric for a Multicast Fast Packet Switch', In *Proc. of Int. Switching Symposium*, Stochkholm, Sweden, May.

NICOLAOU, C. (1990), 'An Architecture for Real Time Multimedia Communication Systems', *IEEE Journal on Selected Areas in Communications*, Vol. 8, No. 3, April, pp. 391–400.

NORROS, I., ROBERTS, J. AND VIRTAMO, J. (1991), 'The Superposition of Variable Bit Rate Sources in An ATM Multiplexer', *IEEE Journal on Selected Areas in Communications*, Vol. 9, No. 3, April, pp. 378–387.

OHNISHI, H., OKADA, T. AND NOGUCHI, K. (1988), 'Flow Control Schemes and Delay/Loss Tradeoff in ATM Network', *IEEE Journal on Selected Areas in Communications*, Vol. 6, No. 9, December, pp. 1609–1616.

OHTA, H. AND KITAMI, T. (1990), 'Simulation Study of the Cell Discard Process and the Effect of Cell Loss Compensation in ATM Networks', *IEICE Transaction on Communication*, Vol. E73-B, No. 10, October, pp. 1704–1711.

OHTA, S., SATO, K. AND TOKIZAWA, I. (1988), 'A Dynamically Controllable ATM Transport Network Based on the Virtual Path Concept', In *Proc. of IEEE GLOBECOM Conference*, Fort Lauderdale, FL, November, pp. 1272–1276.

OKADA, T., OHNISHI, H. AND MORITA, N. (1991), 'Traffic Control in Asynchronous Transfer Mode', *IEEE Transaction on Communications*, Vol. 29, No. 9, September, pp. 58–62.

ONG, L. AND SCHWARTZ, M. (1991), 'Distributed Control of Multiparty, Multimedia Broadband Services', In *Proc. of IEEE ICC Conference*, Denver, CO, June, pp. 847–851.

PANCHA, P. AND EL ZARKI, M. (1993), 'Bandwidth Requirements of Variable Bit Rate MPEG Sources in ATM Networks', In *Proc. of IEEE INFOCOM Conference*, San Francisco, CA, March, pp. 902–909.

PATTAVINA, A. (1993), 'Nonblocking Architectures for ATM Switching', *IEEE Communication Magazine*, Vol. 31, No. 2, February, pp. 38–48.

PATTERSON, J. AND EGIDO, C. (1990), 'Three Keys to the Broadband Future: View of Applications', *IEEE Network Magazine*, Vol. 4, No. 2, March, pp. 41–47.

PAWLIKOWSKI, K. (1990), 'Steady-State Simulation of Queueing Processes: A Survey of Problems and Solutions', *ACM Computing Surveys*, Vol. 22, No. 2, June, pp. 123–170.

PETR, D. AND FROST, V. (1991), 'Nested Threshold Cell Discarding for ATM Overload Control: Optimization Under Cell Loss Constraints', In *Proc. of IEEE INFOCOM Conference*, Bal Harbour, FL, April, pp. 1403–1412.

POUZIN, L. (1981), 'Methods, Tools, and Observations on Flow Control in Packet-Switched Data Networks', *IEEE Transaction on Communications*, Vol. 29, No. 4, April, pp. 413–426.

RAMAMURTHY, G. AND DINGHE, R. (1991), 'Distributed Source Control: A Network Access Control for Integrated Broadband Packet Networks', *IEEE Journal on Selected Areas in Communications*, Vol. 9, No. 7, September, pp. 990–1001.

RASMUSSEN, C. AND SORENSEN, J. (1989), 'A Simple Call Acceptance Procedure in an ATM Network', In *Proc. of ITC Specialist Seminar*, Adelaide, Australia, September.

RASMUSSEN, C., SORENSEN, J., KVOLS, K. AND JACOBSEN, S. (1991), 'Source Independence Call Acceptance in ATM Networks', *IEEE Journal on Selected Areas in Communications*, Vol. 9, No. 3, April, pp. 351–358.

RATHGEB, E. (1991), 'Modeling and Performance Comparison of Policing Mechanisms for ATM Networks', *IEEE Journal on Selected Areas in Communications*, Vol. 9, No. 3, April, pp. 325–334.

RATHGEB, E. (1992), 'Policing of Realistic VBR Video Traffic - A Case Study', In *Broadband Communications - A. Casaca (ed.)*, Elsevier Science Publishers, B.V., North-Holland, pp. 287–300.

REN, J., MARK, J. AND WONG, J. (1994), 'Performance Analysis of a Leaky-Bucket Controlled ATM Multiplexer', *Performance Evaluation*, Vol. 19, No. 1, January, pp. 73–101.

RICE, W. AND SPEARS, D. (1992), 'What's New in B-ISDN Standards?', In *High-Speed Communication Networks - H. Perros (ed.)*, Plenum Press, New York, pp. 125–135.

ROBERTS, J. (1991), 'Traffic Control in the B-ISDN', In *Proc. of the 13th Int. Teletraffic Congress*, Copenhagen, Denmark, June, pp. 80–91.

ROBERTS, J. AND VIRTAMO, J. (1991), 'The Superposition of Periodic Cell Arrival Processes in An ATM Multiplexer', *IEEE Transaction on Communications*, Vol. 39, No. 2, February, pp. 298–303.

ROSSITER, M. (1987), 'A Switched Poisson Model for Data Traffic', *Australian Telecommunication Review*, Vol. 21, No. 1, pp. 53–57.

ROTHERMEL, K. (1990), 'Priority Mechanisms in ATM Networks', In *Proc. of IEEE GLOBECOM Conference*, San Diego, CA, December, pp. 847–851.

RUBIN, I. AND LIN, K. (1993), 'A Burst-Level Adaptive Input-Rate Flow Control Scheme for ATM Networks', In *Proc. of IEEE INFOCOM Conference*, San Francisco, CA, March, pp. 386–394.

SAITO, H. (1992), 'Hybrid Connection Admission Control in ATM Networks', In *Proc. of IEEE ICC Conference*, Chicago, MI, June, pp. 699–703.

SAITO, H., KAWARASAKI, M. AND YAMADA, H. (1991), 'An Analysis of Statistical Multiplexing in An ATM Transport Network', *IEEE Journal on Selected Areas in Communications*, Vol. 9, No. 3, April, pp. 359–367.

SALLBERG, K., STAVENOW, B. AND ANDERSEN, I. (1990), 'A Resource Allocation Framework in B-ISDN', In *Proc. of Int. Switching Symposium*, Stochkholm, Sweden, May.

SCHOUTE, F. (1988), 'Simple Decision Rules for Acceptance of Mixed Traffic Streams', In *Proc. of the 12th Int. Teletraffic Congress*, Torino, Italy, January, pp. 771–777.

SCHWARTZ, M. (1987), *Telecommunication Networks: Protocols, Modeling and Analysis*, Addison Wesley Publishing Company, Reading, MA.

SIDI, M., LIU, W., CIDON, I. AND GOPAL, I. (1989), 'Congestion Control through Input Rate Regulation', In *Proc. of IEEE GLOBECOM Conference*, Dallas, TX, November, pp. 1764–1768.

SOHRABY, K. AND SIDI, M. (1991), 'On the Performance of Bursty and Correlated Sources Subject to Leaky Bucket Rate-Based Access Control Schemes', In *Proc. of IEEE INFOCOM Conference*, Bal Harbour, FL, April, pp. 426–434.

SRIRAM, K. (1990), 'Dynamic Bandwidth Allocation and Congestion Control Schemes for Voice and Data Multiplexing in Wideband Packet Technology', In *Proc. of IEEE ICC Conference*, Atlanta, GA, April, pp. 1003–1009.

SRIRAM, K. AND WHITT, W. (1986), 'Characterizing Superposition Arrival Processes in Packet Multiplexers for Voice and Data', *IEEE Journal on Selected Areas in Communications*, Vol. 4, No. 6, September, pp. 833–846.

SRIRAM, K., MCKINNEY, R. AND SHERIF, M. (1991), 'Voice Packetization and Compression in Broadband ATM Networks', *IEEE Journal on Selected Areas in Communications*, Vol. 9, No. 3, April, pp. 294–304.

STALLINGS, W. (1989), *ISDN: An Introduction*, Macmillan Publishing Company, New York, NY.

STALLINGS, W. (1991), *Data and Computer Communications*, Macmillan Publishing Company, New York, NY.

STEINMETZ, R. (1990), 'Synchronization Properties in Multimedia Systems', *IEEE Journal on Selected Areas in Communications*, Vol. 8, No. 3, April, pp. 401–412.

SUMITA, S. AND OZAWA, T. (1988), 'Achievability of Performance Objectives in ATM Switching Nodes', In *Proc. of the Int. Seminar on Performance of Distributed and Parallel Systems*, Kyoto, Japan, December, pp. 45–56.

SURI, S., TIPPER, D. AND MEEMPAT, G. (1994), 'A Comparative Evaluation of Space Priority Strategies in ATM Networks', In *Proc. of IEEE INFOCOM Conference*, Toronto, Canada, June, pp. 516–523.

SUTHERLAND, S., ADDIE, R. AND BURGIN, J. (1988), 'Information Transfer Protocols for the Broadband ISDN', In *Proc. of the Third Australian Fast Packet Switching Workshop*, Melbourne, Australia, May.

SUZUKI, T. (1994), 'ATM Adaptation Layer Protocol', *IEEE Communication Magazine*, Vol. 32, No. 4, April, pp. 80–83.

TIPPER, D., PAPPU, S., COLLINS, A. AND GEORGE, J. (1993), 'Space Priority Buffer Management for ATM Networks', In *Asynchronous Transfer Mode Networks - Y. Viniotis and R.O. Onvural (eds.)*, Plenum Press.

TOBAGI, F. (1990), 'Fast Packet Switch Architectures for Broadband Integrated Services Digital Networks', *Proceedings of IEEE*, Vol. 78, No. 1, January, pp. 133–167.

TRAN-GIA, P. (1986), 'Discrete-Time Analysis for the Interdeparture Distribution of GI/G/1 Queues', In *Proceeding of Teletraffic Analysis and Computer Performance Evaluation - O.J. Boxma, J.W. Cohen and H.C. Tijms (eds.)*, Elsevier Science Publishers, B.V., North-Holland, pp. 341–357.

TRAN-GIA, P. (1989), 'Analysis of a Load-Driven Overload Control Mechanism in Discrete-Time Domain', In *Teletraffic Science for New Cost-Effective System Networks and Services - M. Bonatti (ed.)*, Elsevier Science Publishers, B.V., North-Holland, pp. 379–385.

TRAN-GIA, P. (1992), 'Analysis of Polling Systems with General Input Process and Finite Capacity', *IEEE Transaction on Communications*, Vol. 40, No. 2, February, pp. 337–344.

TRAN-GIA, P. AND RAITH, T. (1985), 'Multiqueue Systems with Finite Capacity and Non Exhaustive Cyclic Service', In *Proc. Int. Seminar on Computer Networking and Performance Evaluation*, Tokyo, Japan, pp. 213–225.

TURNER, J. (1986), 'New Direction in Communications (or Which Way to the Information Age?)', *IEEE Communication Magazine*, Vol. 24, No. 10, October, pp. 8–15.

TURNER, J. (1988), 'Design of a Broadcast Packet Switching Network', *IEEE Transaction on Communications*, Vol. 36, No. 6, June, pp. 734–743.

TURNER, J. (1992), 'Managing Bandwidth in ATM Networks with Bursty Traffic', *IEEE Network Magazine*, September, pp. 50–58.

UNTEREGELSBACHER, E. AND MOUFTAH, H. (1991), 'PDF Based Congestion Control in ATM Networks', In *Proc. of IEEE ICC Conference*, Denver, CO, June, pp. 211–215.

VAKIL, F. AND SAITO, H. (1991), 'On Congestion Control in ATM Networks', *IEEE Lightwave Communication Systems Magazine*, Vol. 2, No. 3, August, pp. 55–65.

VERBIEST, W. AND PINNOO, L. (1989), 'A Variable Bit Rate Coded for Asynchronous Transfer Mode Networks', *IEEE Journal on Selected Areas in Communications*, Vol. 7, No. 5, May, pp. 761–770.

VERBIEST, W., PINOO, L. AND VOETEN, B. (1988), 'Statistical Multiplexing of Varaible Bit Rate Video Sources in Asynchronous Transfer Mode Networks', In *Proc. of IEEE GLOBECOM Conference*, Fort Lauderdale, FL, November, pp. 208–213.

WANG, S. AND SILVESTER, J. (1993), 'A Fast Performance Mode for Real-Time Multimedia Communication', In *High Speed Networks and Their Performance - H.G. Perros and Y. Viniotis (eds.)*, Elsevier Science Publishers, B.V., North-Holland, pp. 199–216.

WANG, W., SAADAWI, T. AND AIHARA, K. (1990), 'Bandwidth Allocation for ATM Networks', In *Proc. of IEEE ICC Conference*, Atlanta, GA, April, pp. 439–442.

WASEM, O., GOODMAN, D., DVORAK, C. AND PAGE, H. (1988), 'The Effect of Waveform Substitution on the Quality of PCM Packet Communications', *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 36, No. 3, March, pp. 342–348.

WATERMAN, M., HARTANTO, V. AND SIRISENA, H. (1993), 'Voice Traffic Control in B-ISDNs', In *Proc. of Australian Broadband Switching and Services Symposium*, Wollongong, Australia, July, pp. 38–45.

WOODRUFF, G. AND KOSITPAIBOON, R. (1990), 'Multimedia Traffic Management Principles for Guaranteed ATM Network Performance', *IEEE Journal on Selected Areas in Communications*, Vol. 8, No. 3, April, pp. 437–446.

WOODRUFF, G., ROGERS, R. AND RICHARDS, P. (1988), 'A Congestion Control Framework for High-Speed Integrated Packetized Transport', In *Proc. of IEEE GLOBECOM Conference*, Fort Lauderdale, FL, November, pp. 203–207.

YAMANAKA, N., SATO, Y. AND SATO, K. (1993), 'Usage Parameter Control and Bandwidth Allocation Methods Considering Cell Delay Variation in ATM Networks', *IEICE Transaction on Communication*, Vol. E76-B, No. 3, March, pp. 270–279.

YAU, V. AND PAWLIKOWSKI, K. (1992), 'Improved Nested-Threshold-Cell-Discard Buffer Management Mechanisms', In *Proc. of IEEE Region 10 Conference*, Melbourne, Australia, November.

YIN, N. AND HLUCHYJ, M. (1991), 'A Dynamic Rate Control Mechanism for Source Coded Traffic in a Fast Packet Network', *IEEE Journal on Selected Areas in Communications*, Vol. 9, No. 7, September, pp. 1003–1012.

ZHANG, J. (1993), 'Performance Study of Markov Modulated Fluid Flow Models with Priority Traffic', In *Proc. of IEEE INFOCOM Conference*, San Francisco, CA, March, pp. 10–17.

ZHANG, H. AND FERRARI, D. (1993), 'Rate-Controlled Static-Priority Queueing', In *Proc. of IEEE INFOCOM Conference*, San Francisco, CA, March, pp. 227–236.

ZHANG, Y., WU, W., KIM, K., PICKHOLTZ, R. AND RAMASASTRY, J. (1991), 'Variable Bit-Rate Video Transmission in the Broadband ISDN Environment', *Proceedings of IEEE*, Vol. 79, No. 2, February, pp. 214–221.

ZUKERMAN, M., YAO, L., POTTER, P., BUDIKIS, Z. AND BLASIKIEWICZ, M. (1991), 'A Generic Flow Control for B-ISDN', In *Proc. of Australian Broadband Switching and Services Symposium*, Sydney, Australia, July, pp. 118–129.

APPENDIX

DESC++: AN OBJECT-ORIENTED TOOL FOR SIMULATING TELECOMMUNICATION NETWORKS *

Abstract–DESC++ (Discrete Event Simulation package using C++) has been developed as a tool for performance evaluation studies of new communication protocols and architecture of telecommunication networks. The package consists of various object classes that allow discrete event simulation programs of telecommunication networks to be easily developed in a queueing network paradigm. Some additional features in the DESC++ package include event tracing and reporting, input/output redirection and interactive simulation.

We focus on special approaches used in DESC++ for addressing two central issues in performance evaluation studies based on quantitative stochastic simulation, namely (i) development and implementation of simulation models, and (ii) analysis of simulation output data with automatic precision control of the final results. An example is given to illustrate pertinent features of the solutions to these issues implemented in DESC++.

Key words: Discrete event simulation, object-oriented programming, quantitative stochastic simulation, telecommunication networks.

1 INTRODUCTION

In recent years, stochastic discrete event simulation has been widely used in analysing and designing new communication networks as it offers flexibility of performance modelling at any level of detail, restricted only by the computing resources of available computers.

While planning extensive simulation studies of various new solutions for future high-speed telecommunication networks, we realised that simulation model development times could be significantly reduced if the various model components were re-usable. While numerous simulation languages and packages exist [21], they are tailored to specific applications, and so normally do not offer the level of modeling flexibility that is needed for studying research problems of rapidly evolving new telecommunication networks based on Asynchronous Transfer Mode (ATM) or Wave Division Multiplexing (WDM) technology.

*This text has been submitted for publication in Computers and Electrical Engineering (coauthors: K. Pawlikowski, H.R. Sirisena and W. Kreutzer).

On the other hand, while it is obvious that in quantitative simulation studies one should pay special attention to accuracy of the final results of these (simulated) statistical experiments, it is suprising that a few simulation packages are equipped with statistical analysers of the output simulation data, and even fewer are able to control the statistical errors of their final results. This has created an alarming situation in which "... <there is no> other field of engineering or science where similar liberties are taken with empirical data ..." [10]. And, this has continued despite warnings that "... the brute force analysis of complex systems <i.e. by simulation but without statistical analysis of simulation output data> can provide undue confidence in erroneous results ..." [23] , or that "... computer runs yield a mass of data but this mass may turn into a mess <if the random nature of such output data is ignored>, and then, instead of an expensive simulation model, a toss of the coin had better be used..." [16].

Recent achievements in software engineering and in statistics allow us to solve both issues. Effective solution of the former can be achieved by applying the object-oriented programming paradigm which offers a high degree of modularity and re-usability of simulation (sub-)models [2, 14]. Thus, we decided to design our own simulation package in an object-oriented language, specifically C++. It was decided too that the package should be user-friendly not only at the stage of constructing simulation models, but also when simulations are executed. Using our expertise in the area of quantitative stochastic simulation of queueing processes [25], the resulting package, named DESC++ (Discrete-Event Simulation in C++), is able to control the length of simulation experiments automatically, stopping simulation when the confidence intervals of all analysed performance measures become sufficiently narrow. Since users of DESC++ have to declare only the required level of confidence of the final estimates and the required (relative) width of their confidence intervals, the package is planned to be introduced in student laboratories of the Department of Electrical and Electronic Engineering. DESC++, despite of its complexity, was essentially written by one person (the first author) in about three months as a tool to be later used in his Ph.D. project; thus it costs less than a typical commercial simulation package.

For discussing how the above issues were addressed, we structure this paper as follows. In the next section, the approach adopted for developing reusable simulation models is described. This is followed by a discussion of its implementation aspects in Section 3. Section 4 presents an overview of the DESC++ simulation package. An example to illustrate the usage and special features of DESC++ is given in Section 5. Section 6 describes the approaches used for estimating the desired measure of performance from the simulation output data and Section 7 concludes this paper with a discussion of future development of DESC++.

2 DEVELOPING SIMULATION MODEL

The event-scheduling approach and the process-interaction approach are two techniques commonly used for modelling systems. Both of them have found application in currently available simulation software. For example, SIMSCRIPT II.5 [15] and GASP [27] are based on event scheduling, while GPSS [28] and SIMULA [6] use process interaction. Usually, when creating a model of a system, the system's static structure is described using an entity-attribute-set representation [8, 18]. For instance, a system representing a virtual circuit in a packet switching network can be modeled as shown in Figure 1. Packet, source, server and receiver are of type entity, priority is the attribute of a packet, and queue is a set for collecting packet entities.

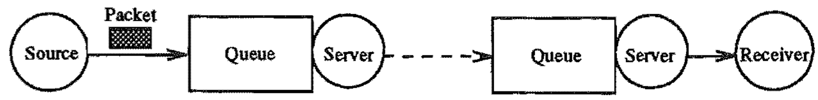


Figure 1 Simulation model for a virtual circuit.

DESC++ caters for both event-oriented and process-oriented approaches and bears a resemblance to DEMOS, a SIMULA context for discrete event simulation [1]. The major difference of DESC++ from DEMOS is in its representation of a system. In DESC++, systems are considered as comprising a number of subsystems. In real world systems, there is a number of possible arrangements of the subsystems. Therefore when defining a subsystem, we require that each subsystem should perform a well-defined function and that it be loosely coupled to other subsystems through a standard interface, as shown in Figure 2. We call such a subsystem a *block with standard interface* or simply a *block*. Standard elements, such as packets or messages, are assumed to flow from one subsystem to another. Within each subsystem, the elements are processed, which results in modifications of the attributes of elements. The standard element is referred to as an *entity*.

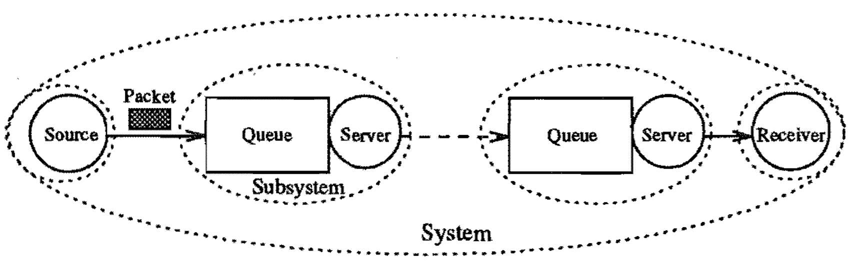


Figure 2 Subsystems for a virtual circuit.

Based on this approach, a system can be described using what we call an *entity-block* representation. Specific features of this representation when compared with the entity-attribute-set representation, is that it distinguishes between minor and major entities, where minor entities have only attributes, while major entities have both attributes and procedures. Minor entities, or simply entities, are assumed to flow from one block to another. A block itself can be viewed as comprising several major entities which have the same functionalities, and their related sets. It can also incorporate other blocks. This offers a form of hierarchical structuring to manage the modelling complexity of large networks. All entities exist only temporarily, while blocks can exist either temporarily or permanently. Temporary existence of a block is manifested by the creation and deletion of the block during a simulation process. A classic example of such a block is a call which originates randomly during a simulation process, generates packets to be sent through a network and terminates after a given holding time.

3 IMPLEMENTING THE MODEL

The entity-block approach represents the object-oriented programming (OOP) paradigm and provides the technology for simple and natural modelling of systems, using *blocks with standard interfaces* as special classes of objects. The modularisation capability of OOP allows the encapsulation of all data and procedures related with a block/subsystem to be incorporated into a software block object, and furthermore hierarchical structuring allows the block object to incorporate other block objects representing subsystems at higher levels of detail. Modularity localises effects of any necessary modification to the model of a subsystem and it allows the developer to concentrate only on the essential details of modification. On the other hand, polymorphism facilitates the provision of standard interfaces between block objects. This allows a block object to be easily substituted by a new block object without the need for modifying other block objects or the whole simulation program.

There are numerous programming languages developed to directly support OOP, such as C++, CLOS, Eiffel, Objective-C, Object-Pascal, SIMULA, and SmallTalk-80. Our choice of C++ as the language in which DESC++ is written was based on its modularity, portability, and fast and efficient compilation. Furthermore C++ has an extensive mathematical library which is very valuable in carrying out statistical analysis of simulation output, as discussed in Section 6. Other benefits of implementing discrete event simulation in C++, including easy extensibility and inherent portability of simulation models, are wider discussed, e.g. in [7] and [14].

The implementation of a simulation model in C++ is through the definition of class Entity and class Block. The following are some essential parts of the definition.

```
class Entity : public Object
{ public:
    static int      class_type;
           char     *name;
};

class Block : public Entity
{ public:
           int      action_type;
           double   event_time;

           Block(char *initName, Block *nextBlock);
    virtual ~Block();
    virtual void    forward(Entity& toForward);
    virtual void    process(Entity& toProcess);
    virtual void    action();
            void    schedule(double delay, int initActionType= 1);
            void    hold(double delay);
            void    repeat(int actionToRepeat = 1);
};
```

Each Entity object must have two unique identifiers: *class_type* to differentiate an entity from one class to one from other class, and *name* to differentiate instantiations of entities of the same class. Each Block object, in addition to the data structures derived from class Entity, has data structures and procedures responsible for flow management. Flow management is at the heart of discrete event simulation as it ensures that a model changes its states in the proper sequence. In DESC++, this can be done either by event management or transaction management.

Event management handles the scheduling of events in the simulation and the execution of the scheduled events. The scheduling of an event can be done through calling either *schedule()*, *hold()* or *repeat()* functions. A scheduled event is represented by the attributes *action_type* and *event_time* within a Block object. All possible events for a Block object are listed within the function *action()*, which can be viewed as an administration function. Further description of event management is given in Section 4.2(a).

Transaction management handles the passing of Entity objects from one Block to another. The *forward()* function passes Entity objects to the next Block object after they are generated or processed. At the next Block object, the *process()* function is used to process the arriving entities. Both functions can be viewed as communication functions.

4 OVERVIEW OF DESC++ SIMULATION PACKAGE

4.1 DESC++ Class Hierarchy

In addition to class Entity and class Block, as mentioned in the previous section, the DESC++ package contains a wide collection of classes for building a simulation model. These classes include queues, data collection objects, resources, random number generators, and data analysis objects. The hierarchy of object classes in the DESC++ simulation library is depicted in Figure 3. Two object classes of interest in this paper are class DoubleList and class DataAnalysis. More information on other object classes in the DESC++ package can be found in [11].

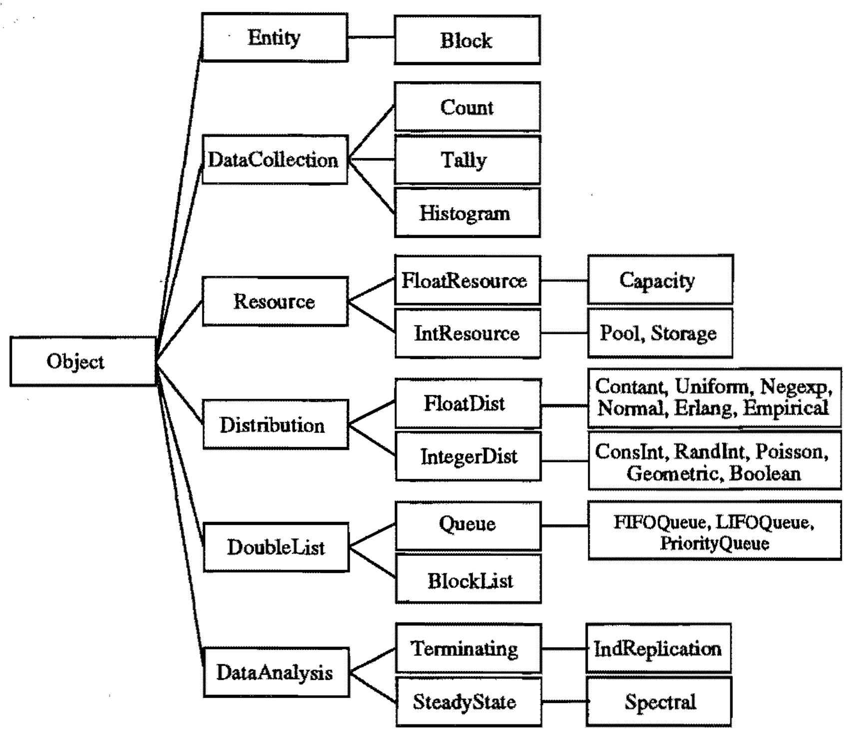


Figure 3 Hierarchy of object classes in DESC++ simulation package.

Class Queue and class BlockList are derived from class DoubleList. Class Queue provides a container for holding various objects derived from class Entity. The container removes the need for an entity to carry any pointer to link itself with other entities in the queue. Such implementation facilitates a better modularisation of Entity and Queue objects. Class BlockList, on the other hand, plays an important role in event management, as will be discussed in the next section.

Class DataAnalysis provides objects, Terminating and SteadyState, used for sequential statistical analysis of simulation output data in the case of *terminating simulation* and *steady-state simulation*, respectively [19]. The details of the methods of sequential statistical data analysis implemented in each type of simulation are presented in Section 6.

4.2 Other DESC++ Facilities

(a) Event Management

In DESC++, there is basically an infinite number of types of event possible, as each block carries its own possible activities within its *action()* function, by using the C++ *switch* statement. When an event is scheduled, the attributes *action_type* and *event_time* within the Block object are set appropriately and the Block object is inserted into a BlockList according to the *event_time* value. Block objects are arranged in a chronological order within the BlockList based on their *event_time* values. A function *nextEvent()* from the class BlockList will detach the Block object at the head of this list and execute the action *action_type* associated with the Block. During the execution of the activity the Block may schedule further events, which causes the Block to be reinserted in the BlockList according to the next *event_time*. The simulation will stop if the BlockList is empty.

The advantage of using a BlockList instead of the usual event list is that it combines both event and block management functionalities. This indicates that any deletion of a Block object will mean the deletion of all activities that may be associated with it. This avoids the need for searching through an event list or event lists (for example early GPSS versions [4] used a two-level concept of a so called "current event chain" and "future event chain" to facilitate event sequencing) for all events associated with the deleted objects. Furthermore using a BlockList appears to facilitate a more natural concept since it allows one to model the location of a Block and the actions that it may do rather than modelling events and the Block involved in the events as with event lists. This is similar to some modern versions of GPSS, e.g. GPSS/H [20], which associate "chains" of transactions (event lists) with blocks in order to speed up event processing.

(b) Reporting and Event Tracing Facilities

The DESC++ package includes *report()*, *trace()*, *snapQueue()*, *snapEventList()*, which can be executed at any time during a simulation process. The function *report*(t_1) produces reports at $t = t_1$, where the report style can be customised. The function *trace*(t_1, t_2) produces traces of a given simulation process for the interval (t_1, t_2), which can be used to verify the simulation model. When a trace during the whole simulation process is required, *trace()* can be used. The functions *snapQueue*(t_1) and *snapEventList*(t_1) list all objects in queues and all previously scheduled events at $t = t_1$, respectively.

(c) Input/Output Redirection Functions

The package also includes *open()/close()*, *switchOpen()/switchClose()* functions for opening or closing primary and secondary files, respectively. The provision of such functions along with *prompt()* and *writeOut()* allows input/output to be read/written to/from a console or file automatically. This avoids the need to differentiate between the *scanf()* and *fscanf()* functions, or the *cout* and *outfile*, where *outfile* is an instantiation of class ofstream of C++.

(d) Interactive Simulation

Interactive simulation is a feature that allows users to stop a simulation, to request a report and an event tracing, and to add or delete block objects interactively during a simulation run.

4.3 Program Format

A simulation program in DESC++ should follow the C++ format. It can be compiled using, e.g. Borland C++ on PC, or GNU G++ on UNIX, and linked to the DESC++ library to produce an executable file. Thus, the main program should be in the following format:

```
#include "desc.h"

[ classes defined by users ]

int main()
{ [ input/output redirection functions, e.g. open("test.out", 'o');]
   initialiseSimulation(t_init);
   [ user model, e.g. newSource = new Source("Source"); ]
   [ simulation control functions, e.g. trace(0.0, 10.0); ]
   startSimulation(t_max);
   [ accessing simulation results; ]
   return 0;
}
```

The function *initialiseSimulation(t_init)* initialises a simulation process by setting the system clock to *t_init*, which is equal to 0.0 by default. This function must be called prior to creating any object instantiations. After the creation of all objects, users can start the simulation process by calling *startSimulation(t_max)* with *t_max* being the maximum simulation length for terminating simulation. For a steady state simulation, *t_max* should be left unspecified. The simulation process will stop when the maximum simulation length is exceeded, or the required statistical precision of all estimates of interest is reached. At the end of the simulation process the package automatically deletes all dynamic objects created during that process. This is very important especially in multiple runs, as otherwise undeleted dynamic objects in one run could interfere with objects in the next run. In the PC version, such interference could cause the computer to run out of memory or to terminate the simulation abnormally, while in the UNIX version it could cause segmentation fault errors.

5 EXAMPLE: VIRTUAL CIRCUIT SIMULATION

To illustrate the usage and some of the features of the DESC++ simulation package, let us consider a sample program written to simulate a virtual circuit in a packet switching network.

The simulation model is based on a tandem of M/M/1 queues. This model allows users to specify at run time the number of nodes to be created in the tandem and the number of sources feeding the virtual circuit. In the simulation program, class Packet is derived from class Entity, while classes FIFONode, Poissonian and Receiver are derived from class Block. The definition for FIFONode and the main program are listed below.

```
class FIFONode : public Block
{ public:
          Queue    *queue;
          Negexp   *service;
          Spectral *delay;
          int       empty;
          FIFONode(char *initName, Block *nextBlock, long int max_obs);
   virtual ~FIFONode() { delete queue; delete service; delete delay; }
   virtual void      process(Entity& toProcess);
   virtual void      action();
};

FIFONode::FIFONode(char *initName, Block *nextBlock, long int max_obs)
  : Block(initName, nextBlock)
{ queue = new Queue("FIFO");
  prompt(&service_rate, "Input the average service rate for %s : ", name);
  service = new Negexp("Service", service_rate);
  delay = new Spectral("Delay", max_obs, 0.95, 0.05);
  empty = 1;
}

void FIFONode::process(Packet& toProcess)
{ ((Packet&)toProcess).arrival_time = _CLOCK;
  queue->addAtTail(toProcess);
  if(empty)
  { schedule(0.0);
    empty = 0;
  }
}

void FIFONode::action()
{ switch(action_type)
  { case 1: Entity *temp = queue->detach();
            delay->update(CLOCK-temp->arrival_time);
            delete temp;
            hold(service->sample());
            break;
    case 2: if(queue->length > 0) repeat();
            else empty = 1;
  }
}

/* Main procedure */
int main()
{ open("mm1.out", 'o');
  initialiseSimulation();
  long int max_obs, seed;
  prompt(&max_obs, "Input the maximum number of observations : ");
                 /* the maximum number of observations that could be collected */
  prompt(&seed, "Input the initial seed : ");
                 /* any positive integer number not greater than 2^31-2. */
  int no_of_nodes, no_of_sources;
  prompt(&no_of_nodes, "Input the number of nodes : ");
  prompt(&no_of_sources, "Input the number of sources : ");
  Dist::setBasicSeed(seed);
```

```
    Receiver *newReceiver = new Receiver("Receiver");
    Block *newNode[10];
    newNode[0] = new FIFONode("Node", newReceiver, max_obs);
    for(int i=1; i<no_of_nodes; i++)
      newNode[i] = new FIFONode("Node", newNode[i-1], max_obs);
    for(int j=0; j<no_of_sources; j++)
      Block *newSource = new Poissonian("Source", newNode[no_of_nodes-1]);
    trace(0.0, 15.0);
    snapEventList(1000.0);
    startSimulation();
    close('o');
    return 0;
}
```

The program does not use any global variables, which demonstrates the advantage of object-oriented modelling styles, since we can encapsulate all attributes and functionalities belonging to a subsystem within a Block object. The encapsulation and standard interface features from a Block object allow users to substitute a FIFONode object for a PriorityNode object, if users want to compare delay performance between FIFO and a priority queueing discipline. In the same way, different classes of sources can be defined to replace the class Poissonian in studying the effect of various source models. As an alternative to defining a new class, users can also modify class FIFONode functionalities to provide priority queueing. The modularity of the DESC++ model allows users to do so without the need to understand or modify any other part of the program.

In the program, we can also notice that the function *startSimulation()* passes no arguments, which indicates a steady-state simulation. This leaves the length of simulation to be decided by the Spectral objects, created within each FIFONode object. The objects are responsible for detecting stopping conditions of simulation and stop the simulation when the required precision for all performance measures analysed is obtained or the maximum number of observations *max_obs* is reached. Various Spectral objects can be created independently within each Block object for each performance measure analysed. The statistical procedures used in the objects are discussed in Section 6.

As a sample run, we consider a virtual circuit with two nodes. The virtual circuit can be modelled by an arrangement of Source1-Node2-Node1-Receiver1. The average service rate per node is assumed to be 10/7=1.429. A source is assumed to generate packets at an average rate of 1.0 packets per second, resulting in a utilisation of 0.7. An input sample and a truncated output listing of such virtual circuit simulation is given in the following:

```
Input:
  >> Input the maximum number of observations : 1000000
  >> Input the initial seed : 907
  >> Input the number of nodes : 2
  >> Input the number of sources : 1
  >> Input the average service rate for Node1 : 1.429
  >> Input the average service rate for Node2 : 1.429
  >> Input the average packet generation rate for Source1 : 1.0

Output:
  *** Initialising simulation ...
  *** Simulation commences at 0.000 ...
  *** Tracing commences at 0.000 ...
```

```
At    0.000 – Source1    generates Packet1.
              Node2      inserts Packet1 into queue FIFO2.
                         schedules Node2 now.
              Source1    holds for 0.613 until 0.613.
              Node2      coopts Packet1 from queue FIFO2.
                         holds for 0.242 until 0.242.
At    0.242 –            releases coopt entity Packet1.
              Node1      inserts Packet1 into queue FIFO1.
                         schedules Node1 now.
                         coopts Packet1 from queue FIFO1.
                         holds for 1.860 until 2.101.
                                   :
                                   :
At   14.705 – Node2      releases coopt entity Packet11.
              Node1      inserts Packet11 into queue FIFO1.
                         schedules Node1 now.
                         coopts Packet11 from queue FIFO1.
                         holds for 0.484 until 15.189.
              Node2      coopts Packet12 from queue FIFO2.
                         holds for 1.552 until 16.256.

*** Tracing is switched off at 15.000 ...

*** Future event list at 1000.000 ...
/* Events awaiting execution */
NO. EVENT NAME    EVENT TIME ACTION
 1  Node2          1000.071    2
 2  Node1          1000.159    2
 3  Source1        1000.632    1

*** Simulation is stopped at 2.32e+05 ...
*** Simulation report at 2.32e+05 ...
/* Node2: FIFONODE */
/* Delay: SPECTRAL */
transient = 176; steady = 77207; mean = 1.59; eps = 0.041
    /* where transient = the number of observations collected during the initial transient period;
       steady = the number of observations required for stopping steady state analysis with the
       required precision; mean = the estimated value; eps = the precision of the estimate. */
/* Receiver1: RECEIVER */
/* Delay: SPECTRAL */
transient = 351; steady = 30654; mean = 4.58; eps = 0.036
```

The output shows typical event tracing, event list snapshot and report provided by DESC++. The style of the report has been customised for the objects. The report shows the estimation of average queueing delay at Node2 and end-to-end transfer delay at Receiver1 at the 95% confidence level with a relative precision $eps \leq 0.05$. The relative precision of a result is defined as the ratio of the current half-width of its confidence interval and the current value of its point estimate [19, 25]. The method used for obtaining the results is based on Spectral Analysis [13, 25]. The mean transfer delay at Receiver1 has reached its precise steady state value after 30654 observations. However, the program was not stopped until the mean queueing delay at Node2 reached its precise steady state value, and that happened after 77207 observations were collected. This illustrates the advantage of DESC++ modelling which allows estimations of various performance measures to be carried out independently, while the simulation is stopped only if all the measures reach their steady state or the maximum number of observations is exceeded. Statistical analysis of simulation output data is discussed further in the next section.

6 STATISTICAL ANALYSIS OF SIMULATION OUTPUT

Recognising that output data of quantitative stochastic simulation experiments always have to be properly statistically analysed, one can apply either a fixed-sample-size or an adaptive-sample-size approach [19]. In the former approach, data are analysed after simulation; thus the length of simulation has to be decided before the simulation is started. A post-simulation processor can help with applying sophisticated statistical techniques needed for drawing correct conclusions from collected data [3], but the main problem of how to determine the simulation run length for obtaining accurate final results before simulation is started, without involving users into time-consuming simulation trials, remains unanswered. This inherent difficulty of determining the required length of simulation runs before it is executed has led to publications where results are given without their precision or, as e.g. in [32], where estimates of performance measures are reported with their precision as long as the precision is better than 35% (!). Using a long simulation run may not always solve the problem either.

Station	Relative precision (ϵ)			
	$\rho = 20\%$	$\rho = 60\%$	$\rho = 80\%$	$\rho = 95\%$
1	0.0897	0.0484	0.0479	0.1015
2	0.0589	0.0469	0.0428	0.0904
3	0.1032	0.0549	0.0668	0.1198
4	0.1308	0.0570	0.0487	0.1008
5	0.1103	0.0376	0.0518	0.1374
6	0.1094	0.0729	0.0452	0.0845
7	0.1308	0.0491	0.0601	0.1285
8	0.0688	0.0557	0.0781	0.1187
9	0.0813	0.0800	0.0514	0.1076
10	0.1354	0.0764	0.0625	0.1567
11	0.2085	0.0617	0.1070	0.1723
12	0.1863	0.0908	0.0956	0.2123
13	0.2262	0.0915	0.1068	0.1446
15	-	0.1022	0.1004	0.2025
16	-	0.1448	0.2227	0.2029
17	-	0.2110	0.1261	0.2430
18	-	0.2480	0.1775	0.4306
19	-	-	-	-

Table 1 Simulation of DQDB MAN, 20 stations, run length 2,000,000 time slots.

For example, using a run length of 2 million time slots in simulating a DQDB metropolitan area network with 20 stations (Table 1), it was found that while the mean delay of packets at some stations has reached a reasonable precision (e.g. below 5%) at the required 0.95 confidence level, there were also stations for which the precision of estimates still remains at levels exceeding 30% [22]. While the fixed-sample-size scenario would simply require one to try again with an increased run length, there is no guarantee that the next length selected will be satisfactory.

These problems lead to the conclusion that an adaptive-sample-size approach is the answer. It enables the simulation to be stopped when the results reach a required level of accuracy, following sequential analysis of output data during simulation. In practice, the accuracy of an estimate is measured by its relative precision defined as the ratio of the current half-width of confidence interval (at an assumed confidence level) and the current value of the estimate of the performance measure analysed.

In DESC++ adaptive-sample-size approach is implemented both for terminating and steady-state stochastic simulations of telecommunication networks. The former means that the performance of a network is studied within a well-specified time interval $[t_1, t_2]$, where t_1 and t_2 are starting and terminating points of the simulation, respectively. In this scenario, the well-known method of Independent Replications (IR) provides necessary mechanisms for collecting and analysing statistically independent data. In a fully automated version of IR the simulation length is determined sequentially by analysing the precision of estimates after each consecutive simulation run (replication) is executed. If the results obtained during N replications are averages $\bar{X}_1, \bar{X}_2, \ldots, \bar{X}_N$, then the overall estimator and its precision are given as

$$\bar{X}(N) = \sum_{i=1}^{N} \frac{\bar{X}_i}{N}$$

$$\epsilon = t_{N-1,1-\alpha/2}\sigma\bar{X}(N)$$

where

$$\sigma^2 = \sum_{i=1}^{N} \frac{(\bar{X}_i - \bar{X}(N))^2}{N(N-1)}$$

is the (unbiased) estimator of the variance of $\bar{X}(N)$ and $t_{N-1,1-\alpha/2}$ is the upper $(1-\alpha/2)$ critical point obtained from the Student t-distribution with $(N-1)$ degrees of freedom.

The procedure is implemented in DESC++ as an *IndReplication* object and users can create the object by calling

```
IndReplication *delay = new IndReplication("Delay", N_min, N_max, ci_level, precision);
```

Recording observations is accomplished by calling the procedure

```
delay->update(obs_value);
```

The precision ϵ is calculated for the first time after N_{min} replications. If $\epsilon > \epsilon_{desired}$, then extra replication is needed, after which ϵ is computed again. The process is repeated until $\epsilon \leq \epsilon_{desired}$ or until a specific maximum number of replications N_{max} has been reached. In the latter case it would mean that the required precision of the final results can not be reached within the available time.

Steady-state simulation is used for studying the long-run behaviour of simulated systems, which in the case of stable stochastic processes means analysis of their behaviour in the steady state, known also as statistical equilibrium, theoretically entered by such processes after an infinitely long period of time. In such a case, the only way of implementing adaptive-sample-size is to execute one long simulation run, since IR would require not only the number of replications be pre-determined, but also the replication length were known in advance. A survey of methods for collecting output data and their statistical analysis during single simulation runs is given in [25]. Among these methods, Regenerative Simulation (RS) is often advocated for avoiding problems with the analysis of correlated data. However, we found RS too restrictive for the expected applications of DESC++, since it requires proper, and process-dependent, selection of regenerative points. Additionally, RS cannot be easily applied when simulating heavily loaded networks since in such environments lengths of regenerative cycles can increase to a degree that makes the length of any simulation experiment prohibitively long. The method we implemented in DESC++ is based on a version of Spectral Analysis as proposed by Heidelberger and Welch (SA/HW) [13, 25], and was selected on the basis of extensive investigations of various options [5, 26]. Since SA/HW requires that observations collected during the initial transient period be discarded, DESC++ first determines the length of initial transient period. This process has also been fully automated, see Figure 4.

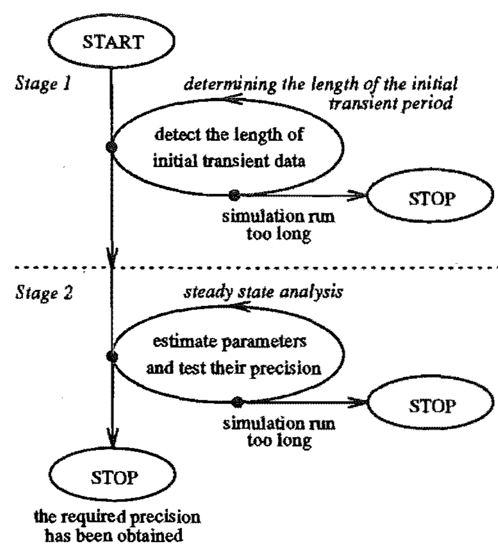


Figure 4 Two-stage sequential methods for statistical analysis in steady-state simulation.

As suggested in [25], the first (rough) estimate of the length of initial transient period is obtained by applying a heuristic rule of thumb that assumes the initial transient period is over when the current value of the estimate becomes statistically representative; see rule R5 of [25]. Then, the search for the beginning of steady-state, indicated by the instant of time when the sequence of collected observations becomes stationary, is continued sequentially by applying a sequential version of the statistical test for testing stationarity of time series, proposed in [29] and adopted for sequential simulation in [25]. When DESC++ decides that the initial transient period is over, all observations that have been collected during that period are discarded and the next stage of simulation begins (Figure 4), during which newly collected observations are analysed at consecutive checkpoints according to SA/HW. Locations of consecutive checkpoints are automatically determined, and analysis of a given performance measure ceases when its precision reaches the required level of precision or a maximum number of observations has been reached. If more than one performance measure is analysed, the simulation is continued until estimates of all performance measures reach the required level of precision. This strategy has been implemented as a class Spectral object in DESC++. An instantiation of a Spectral object can be requested through the use of C++'s *new* statement

```
Spectral *delay = new Spectral("Delay", max_obs, ci_level, precision);
```

Recording observations is accomplished by calling the procedure

```
delay->update(obs_value);
```

To show the problem that one has to face when trying to apply the fixed-sample-size scenario in steady state simulation, and to illustrate the application of DESC++ in the adaptive-sample-size scenario for solving the problem, let us consider simulation studies for obtaining the *average queueing delay* of M/M/1 queues with FIFO and priority queueing disciplines. Let the average arrival rate $\lambda = 1.0$ and average service rate of $\mu = 10/9$, resulting in utilisation $\rho = 0.9$. With priority queueing, we assume that the ratio of high priority traffic level to the overall traffic level is 0.2. With the fixed-sample-size approach, the simulation is replicated 10 times with 10000 observations being collected in each replication. The first replication always starts with a new seed, while the next replication uses the last pseudorandom number drawn in the previous replication as its seed. The initial 500 observations from the beginning of each replication are ignored as likely representing the initial transient period. The final estimated average queueing delay was obtained by averaging the estimated queueing delay from all replications. We carried out 100 experiments of such simulation with different starting seeds. The smallest and the largest final estimated average queueing delay from 100 experiments are shown in Table 2. The exact theoretical results are also given [17].

		Fixed-Sample-Size		Adaptive-Sample-Size	
	Exact	Smallest	Largest	Smallest	Largest
FIFO	8.100	$6.916 \pm 9.48\%$	$9.575 \pm 14.84\%$	$7.640 \pm 4.98\%$	$8.570 \pm 4.62\%$
Priority (high)	0.988	$0.947 \pm 3.16\%$	$1.025 \pm 2.66\%$	$0.939 \pm 3.46\%$	$1.068 \pm 4.76\%$
Priority (low)	9.878	$8.392 \pm 7.66\%$	$11.710 \pm 15.43\%$	$9.355 \pm 4.90\%$	$10.730 \pm 3.96\%$

Table 2 Queueing delay from M/M/1 simulations.

If we look at the ranges of the results obtained under the fixed-sample-size scenario, one can notice the randomness of these results. For example, it is possible that one could obtain such values as 8.392, 0.947 and 9.575 as estimates of mean delay for low priority packets, high priority packets and for the FIFO case. As a result, one could wrongly conclude that packets of both priorities would be better served when priority classes are introduced than in non-priority case. This would obviously contradict the work-conservation law [17]. This indicates that choosing the right number of replications and the right number of observations per replication are crucial for obtaining the right results and hence for drawing the proper conclusions.

With SA/HW, applying the adaptive-sample-size approach, one only has to declare the required level of precision, say 5%, at a 0.95 confidence level. The results (see Table 2) show that the procedure offers a smaller range of estimated values than those under the fixed-sample-size approach with all results having precision below 5%. We can notice that there is no overlap between the range of final estimated values for the FIFO and for the low priority packets, so the danger of drawing a wrong conclusion is diminished.

Looking at the number of observations required to obtain the final estimated values for the FIFO using the SA/HW procedure (see Table 3), we can see that to obtain the required precision of

	Transient	No. of obs.
FIFO	593	421,400
Priority (high)	94	15,560
Priority (low)	566	319,400

Table 3 The average length of transient periods and the total simulation lengths for Spectral Analysis.

the final estimates, on average more than 421,400 observations are required. Furthermore, as one can see from Table 3, in the case of priority queueing, the final precision of the average delay for high priority packets was reached after 15,560 observations, however the simulation kept running until 319,400 observations were collected, i.e. until the required precision for average delay of low priority packets was satisfied as well. The same happens when detecting the initial transient. It is observed that the lengths of the initial transient periods in both cases were different, i.e. 94 observations in high priority packet delay estimation and 566 observations in the low priority one.

The ability to decide the length of initial transient period and the overall simulation period automatically for a number of desired performance measures within the same simulation process is one of the main advantages of the DESC++ implementation of Spectral Analysis.

7 DISCUSSION

In this paper we have described the DESC++ simulation package developed in the Department of Electrical and Electronic Engineering of the University of Canterbury. The DESC++ package offers user-friendly flexibility for constructing simulation models in the object-oriented programming paradigm and automated sequential analysis of simulation output data, which play a crucial role in simulation studies of telecommunication networks. DESC++ has already proved its usefulness in our performance evaluation studies of Asynchronous Transfer Mode (ATM) networks [12].

The vast amount of research on ATM networks has resulted in a tremendous range of ATM network components, such as various policing mechanisms either based on leaky bucket or window-based techniques. Comparative studies of ATM networks could be easily accomplished using the block substitution approach coupled with a large number of basic object classes as provided by DESC++, where the various block modules for ATM networks, such as traffic sources, multiplexers and policing mechanisms, can be derived from the existing class Block, see Figure 5.

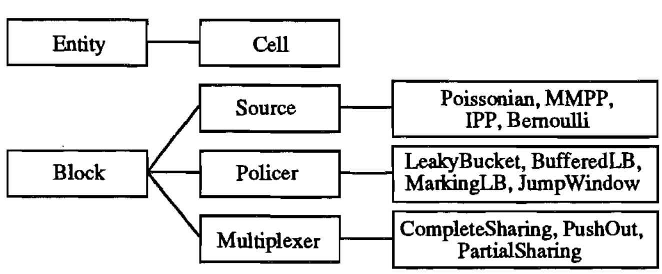


Figure 5 Hierarchy of object classes in ATM network simulation library.

The complexity of ATM networks often prevents mathematical analysis. Having no exact theoretical results as a reference, it is really imperative for simulation results to be properly analysed to avoid drawing incorrect or misleading conclusions about a system's performance. For this purpose a sequential procedure for controlling the precision of simulation results, such as Spectral Analysis, becomes indispensable and the implementation of such techniques in DESC++ has proved to be invaluable for carrying out research in this challenging area. We believe that it should also be advocated for research studies involving stochastic simulation in other area of electrical engineering.

DESC++ has been used in the study of performance of leaky bucket schemes [31] and other traffic management schemes in ATM networks [12]. Works to provide a graphical user interface (GUI) for DESC++ is under way. Further extensions are envisaged to provide the necessary speed-up of the simulation process in the light of the low cell loss probability requirements in future ATM networks by implementing proper techniques for simulating rare events, such as importance sampling [24, 30, 9].

REFERENCES

1. G.M. Birtwistle. *DEMOS: A System for Discrete Event Modelling on Simula.* Macmillan Publishers Ltd., London, 1979.
2. D.P. Bischak and S.D. Roberts. Object-Oriented Simulation. In *Proc. of the 1991 Winter Simulation Conference*, pages 194–203, IEEE Press, 1992.
3. W. Blaisdell and J. Haddock. SIMSTAT: a Tool for Simultion Analysis. In *Proc. of the 1992 Winter Simulation Conf.*, pages 421–425, IEEE Press, 1993.
4. P.A. Bobillier, B.C. Kahan, and A.R. Probst. *Simulation with GPSS and GPSSV.* Prentice-Hall, Inc., Englewood Cliffs, NJ., 1976.
5. K. Chidambaram. Performance Modeling and Evaluation of High Speed Networks, Master's thesis, Department of Electrical and Electronic Engineering, University of Canterbury, New Zealand, 1990.
6. O. Dahl and K. Nygaard. SIMULA: An ALGOL-Based Simulation Language. *Communications of the ACM*, 9(9):671–678, September 1966.
7. D.L. Eldredge, J.D. Mcgregor, and M.K. Summers. Applying the Object-Oriented Paradigm to Discrete Event Simulations Using the C++ Language. *Simulation Journal*, 55(2):83–91, February 1990.
8. G.S. Fishman. *Principles of Discrete-Event Simulation.* John Wiley and Sons, Inc., New York, NY, 1979.
9. M.R. Frater. Fast Simulation of Cell Loss Rates in ATM Using Autoregressive Models. In *Proc. of the Seventh Australian Teletraffic Research Seminar*, pages 201–207, Mannum, Australia, November 1992.
10. B. Gaither. Empty Empiricism. *Performance Evaluation Review*, 18(2):2–3, August 1990.
11. V.F. Hartanto. DESC++: Discrete Event Simulation Library using C++ Language. *Technical report*, University of Canterbury, New Zealand, January, 1993.
12. V.F. Hartanto. *User-Network Oriented Call Control and Traffic Management in B-ISDNs.* PhD thesis, Department of Electrical and Electronic Engineering, University of Canterbury, New Zealand, 1994.

13. P. Heidelberger and P.D. Welch. A Spectral Method for Confidence Interval Generation and Run Control in Simulations. *Communications of the ACM*, 24(4):233–245, April 1981.

14. J.A. Joinse, K.A. Powell, and S.D. Roberts. Object-Oriented Modeling and Simulation with C++. In *Proc. of the 1992 Winter Simulation Conference*, pages 145–153, IEEE Press, 1993.

15. P.J. Kiviat, R. Villaneuva, and H.M. Markowitz. *SIMSCRIPT II.5 Programming Language*. Prentice-Hall, Inc., Englewood Cliffs, NJ., 1973.

16. J.P.C. Kleijnen. The Role of Statistical Methodology in Simulation. In *Methodology in Systems Modelling and Simulation - B.P. Zeigler et al. (eds)*, pages 425–445, Elsevier Science Publishers, B.V., North-Holland, 1979.

17. L. Kleinrock. *Queueing System, Vol. 2: Computer Applications*. John Wiley and Sons, Inc., New York, NY, 1976.

18. W. Kreutzer. *System Simulation Programming Styles and Languages*. Addison Wesley Publishing Company, Reading, MA., 1986.

19. A.M. Law. Statistical Analysis of Simulation Output Data. *Operation Research*, 31(6):983–1029, November 1983.

20. A.M. Law and W.D. Kelton. *Simulation Modeling and Analysis*. McGraw-Hill, Inc., New York, NY., 1991.

21. A.M. Law and M.G. McComas. Simulation Software for Communication Networks: the State of the Art. *IEEE Communication Magazine*, 32(3):44–51, March 1994.

22. Q.Y. Lee. Performance Evaluation of Enhancements to Distributed Queue Dual Bus Protocol, Master's thesis, Department of Electrical and Electronic Engineering, University of Canterbury, New Zealand, 1991.

23. A.M. Marsan, G. Balbo, G. Bruno, and F. Neri. TOPNET: a Tool for the Visual Simulation of Communication Networks. *IEEE Journal on Selected Areas in Communications*, 8(9):1735–1746, December 1990.

24. S. Parekh and J. Warland. A Quick Simulation of Excessive Backlogs in Networks of Queues. *IEEE Transaction of Automatic Control*, 34(1):54–66, January 1989.

25. K. Pawlikowski. Steady-State Simulation of Queueing Processes: A Survey of Problems and Solutions. *ACM Computing Surveys*, 22(2):123–170, June 1990.

26. K. Pawlikowski and V. Yau. Independent Replications versus Spectral Analysis of Output Data in Steady-State Simulation of High Speed Data Networks. In *Proc. of the Sixth Australian Teletraffic Research Seminar*, pages 322–330, Wollongong, Australia, November 1991.

27. A.B.B. Pritsker. *The GASP Simulation Language*. John Wiley and Sons, Inc., New York, NY, 1974.

28. T.J. Schriber. *Simulation Using GPSS*. John Wiley and Sons, Inc., New York, NY, 1974.

29. L.W. Schruben. Control of Initialization Bias in Multivariate Simulation Response. *Communications of the ACM*, 24(4):246–252, April 1981.

30. Q.L. Wang and V.S. Frost. Efficient Estimation of Cell Blocking Probability for ATM Systems. In *Proc. of IEEE ICC Conference*, pages 385–390, Denver, CO, June 1991.

31. M.G. Waterman, V.F. Hartanto, and H.R. Sirisena. Voice Traffic Control in B-ISDNs. In *Proc. of Australian Broadband Switching and Services Symposium*, pages 38–45, Wollongong, Australia, July 1993.

32. M. Zafirovic-Vukotic, I.G. Niemegeers, and D.S. Valk. Performance Analysis of Slotted Ring Protocols in HSLAN's. *IEEE Journal on Selected Areas in Communications*, 6(6):1011–1024, July 1988.